

# UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO DE FIN DE MÁSTER.

## DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE ESTRATEGIAS COLABORATIVAS ENTRE ROBOTS MÓVILES PROVISTOS DE SENSORES DE OLOR INTELIGENTES PARA LOCALIZAR Y POSICIONAR FUENTES DE OLORES CARACTERÍSTICAS.

Máster Universitario en Investigación e Innovación en Tecnologías  
de la Información y las Comunicaciones.

Julio Francisco Acosta Núñez

Octubre 2013



DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE  
ESTRATEGIAS COLABORATIVAS ENTRE ROBOTS  
MÓVILES PROVISTOS DE SENSORES DE OLOR  
INTELIGENTES PARA LOCALIZAR Y POSICIONAR  
FUENTES DE OLORES CARACTERÍSTICAS.

AUTOR: Julio Francisco Acosta Núñez

DIRECTOR: Dr. Guillermo González de Rivera Peces

Grupo de Investigación HCTLab

Dpto. de Tecnología Electrónica y de las Comunicaciones  
(en colaboración con el GNB, del Dpto. de Ingeniería Informática)

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Octubre 2013

# Resumen

## Resumen.

En el presente trabajo de fin de máster se ha implementado, a partir de la plataforma mecánica de robot desarrollada en [6], un nuevo robot que tiene como objetivo fundamental el localizar una fuente de un olor característico, y transmitir de forma colaborativa su posición a los demás robots que se encuentren realizando la misma tarea, y que sean parte de una red de robots inalámbricos que ejecuten tareas específicas en un entorno común.

Para ello se han incorporado al robot nuevas características operativas, como son: en primer lugar dotarle de una nariz electrónica inteligente que detecta la concentración del odorante en el ambiente, la concentración de mayor gradiente, informa al robot la dirección a la cual debe dirigirse, esta nariz inteligente ha sido desarrollada en el grupo de Neurocomputación Biológica, y es un dispositivo que entrega una señal digital acondicionada como protocolo de comunicaciones digitales I<sup>2</sup>C, procesando esta información el robot se posicionará y dirigirá en dirección de la fuente de olor, confirmando periódicamente su posición con respecto a la mayor concentración de olor que paulatinamente va detectando. Al mismo tiempo informa de la posición en la que ha descubierto la fuente de olor a los demás robots localizados en su mismo entorno, de forma que ellos también corrijan, de ser el caso, sus trayectorias y se dirijan hacia la fuente de olor; el mismo robot, tiene también la función de comparar su lectura de la posición de la fuente de olor y compararla con las lecturas de los demás robots, y de esta forma llegar más rápidamente a la determinación del sitio exacto en donde se encuentra el odorante, y concentrarse todos los robots en él.

Los medios eléctricos de los que se sirve para realizar tal tarea son: una comunicación por RF entre todos los robots mallados en una red inalámbrica, que utiliza módulos XBee para la comunicación entre robots y una estación central que coordina dicha red. La parte motriz del robot se ha llevado a cabo mediante dos motores de cc que comandan dos ruedas y que determinan su movimiento mediante dos pares de encoders situados en los ejes de las ruedas. Ambas señales tanto de los encoders, como de los *drivers* de los motores, provienen desde y hacia el controlador del robot.

El Dispositivo que gobierna por completo al robot es una tarjeta con un microprocesador, denominada Beaglebone, que posee un sistema operativo embebido, en este caso Linux Ubuntu, y que posee elementos versátiles de comunicación y control, y acepta varios lenguajes de programación como C++, Java Script y Phyton.

## **Palabras Clave**

Nariz electrónica inteligente, Red inalámbrica de Robots móviles, Trabajo colaborativo entre robots.

## **Abstract.**

In this paper, master's thesis has been implemented, from the robot mechanical platform developed in [6 ], a new robot that fundamental aims locate a source of a characteristic odor and collaboratively transmit its position the other robots that are performing the same task, and are part of a network of wireless robots running specific tasks in a common environment .

This has been incorporated into the new robot operating characteristics, such as: first endow an intelligent electronic nose that detects the concentration of the odorant in the air, the higher concentration gradient, tells the robot which direction to go, this smart nose has been developed in the group Biological Neurocomputation, and is a device that delivers a digital signal refurbished as I<sup>2</sup>C digital communications protocol, processing this information, the robot will be positioned and directed away from the odor source, periodically confirming position with respect to the largest concentration of odor is gradually detected. At the same time reports the position in which he found the odor source located to the other robots in the same environment, so that they also correct, if applicable, their careers and are directed toward the odor source, the same robot, has also the function of comparing the reading of the position of the odor source and comparing the readings of the other robots, and thus more rapidly reach the determination of the exact site where the odorant, and concentrate all robots in it.

Electrical means which are used to perform this task are: RF communication among all meshes robots in a wireless network, which uses XBee for communication between robots and a central station that coordinates the network. The driving part of the robot is carried out using two DC motors which command two wheels and which determine its movement by two pairs of encoders mounted on the wheel axles. Therefore both signals of the encoders, as the drivers of the motors comes from and to the robot controller.

The Device controlling completely the robot is a card with a microprocessor, called BEAGLEBONE, which has an embedded operating system, in this case Linux Ubuntu, and possesses versatile elements of communication and control, and supports several programming languages like C++, JavaScript and Python.

## **Keywords**

Intelligent Electronic Nose, Mobile robots wireless network, collaborative work between robots.



## **Agradecimientos**

Al Espíritu Santo, por tener misericordia de esta alma y derramar tantos dones inmerecidos.

A Guillermo González de Rivera Peces y Javier Garrido, por su calidad humana,

A todos esos Chicos que trabajan en HCTLab y no mezquinan una sonrisa: Luis, Nasib, Pablo, Pedro, Mario y principalmente Carlos.

Especialmente a Ariana, Alejandro y Lucy, no solo este pequeño trabajo es para ustedes, todo mi gran esfuerzo y mi vida entera.

**Julio Acosta.**

**Madrid 2013.**



# Índice de Contenidos

1.	Introducción.	19
1.1	Motivación.	19
1.2	Objetivos.	21
1.3	Organización de la Memoria.	21
2	Estado del arte.	27
2.1	NARICES ARTIFICIALES, GENERALIDADES Y EVOLUCIÓN.	27
2.1.1	PRINCIPIOS Y CARACTERÍSTICAS FUNDAMENTALES DE NARICES ARTIFICIALES:	27
2.1.1.1	Nariz Electrónica Artificial.	28
2.1.1.2	Biomarcadores Volátiles.	28
2.1.1.3	Biosensores.	29
2.1.1.4	Electroquímicos.	29
2.1.1.5	Ópticos.	29
2.1.1.6	Dispositivos piezoeléctricos basados en ondas acústicas.	29
2.1.1.7	Procesamiento de señales.	29
2.1.2	APLICACIONES DE LAS NARICES ELECTRONICAS.	30
2.1.2.1	Desarrollo conceptual de la nariz electrónica.	32
2.1.2.2	Instrumentación de una Nariz Electrónica.	33
2.1.2.3	Análisis de la información de las narices electrónicas.	33
2.2	APLICACIONES DE NARICES ELECTRONICAS MONTADAS SOBRE ROBOTS.	34
2.2.1	Tecnologías de detección y movimiento de robots.	35
2.2.2	Estructura del robot de olfato.	35
2.2.3	Distribución del campo de olor.	36
2.2.4	Rastreo de la fuente de olor.	37
2.2.4.1	Estrategia de búsqueda.	37
2.2.4.2	Control de movimiento basado en la partida.	38
2.2.5	Navegación en robots móviles basados en olfato.	38
2.2.6	Arquitectura de control del robot.	39
3	Descripción General del Sistema.	43
3.1	Descripción general del robot.	43
4	Diseño e implementación de la Plataforma Robótica.	49

4.1	Placa Beaglebone. ....	50
4.2	Beaglebone características Generales. ....	52
4.2.1	Procesador: .....	52
4.2.2	Memoria.....	53
4.2.3	Interfase USB.....	53
4.2.4	Puerto serial. ....	53
4.2.5	Puerto JTAG.....	53
4.2.6	Puerto USB0. ....	53
4.2.7	Conector Micro SD. ....	53
4.2.8	Puerto USB1. ....	54
4.2.9	Puerto USB Cliente. ....	54
4.2.10	Fuentes de poder. ....	54
4.2.11	Botón Reset. ....	54
4.2.12	Indicadores de estado. ....	54
4.2.13	Entrada CTI JTAG. ....	55
4.2.14	Interfases de expansión. ....	55
4.2.15	LCD.....	55
4.2.16	Puerto GPMC.....	55
4.2.17	MMC1.....	55
4.2.18	SPI.....	55
4.2.19	I <sup>2</sup> C. ....	56
4.2.20	Puerto Serial.....	56
4.2.21	Conversores A/D. ....	56
4.2.22	GPIO.....	56
4.2.23	CAN BUS. ....	56
4.2.24	TIMERS. ....	56
4.2.25	PWM.....	56
4.3	Método de desarrollo de software en la Beaglebone.....	58
4.4	Comunicaciones y puertos de expansión de la placa Beaglebone.....	62
4.5	Placa Adaptadora. ....	72
4.6	Diseño de la Placa.....	74
4.7	Fuente de alimentación. ....	76

4.8	Motores y encoders. ....	78
4.9	Conexión de la Nariz electrónica.....	82
4.10	Sistema de comunicaciones .....	84
4.11	Uso de recursos de Beaglebone.....	85
4.12	Construcción del Hardware relacionado.....	88
4.13	Construcción del chasis y procedimientos de acoplamiento y operación. ....	91
4.14	Hardware destinado a los sensores de obstáculos. ....	95
4.15	Integración de la nariz electrónica.....	97
4.16	Incorporación del sistema de comunicaciones. ....	101
5	Implementación del Sistema de detección de Olor. ....	107
5.1	En la estrategia de búsqueda. ....	107
6	Implementación del sistema de comunicaciones y la colaboración entre agentes.....	121
6.1	Tipos de dispositivos configurables en una red con XBee. ....	122
6.1.1	El Coordinador.....	122
6.1.2	Los Routers.....	122
6.1.3	End Device.....	123
6.1.4	Aplicaciones.....	123
6.1.5	Modos de Operación.....	124
6.1.5.1	Modo RECIBIR/TRANSMITIR.....	125
6.1.5.2	Modo de Bajo Consumo (Sleep Mode). ....	125
6.1.5.3	Modo de Comando.....	125
6.1.5.4	Modo Transparente .....	126
6.1.5.5	Modo de operación API.....	126
6.1.5.6	Idle.....	127
6.2	Configuración de los módulos XBee.....	127
6.3	Implementación del hardware y software del sistema XBee.....	132
6.4	Sistema de supervisión.....	135
7	Integración, pruebas y resultados.....	143
7.1	Integración del hardware del sistema y desarrollo del software.....	143
7.2	Funciones software de los distintos niveles.....	144
7.3	Pruebas y resultados.....	151
7.3.1	Prueba de motores.....	151

7.3.2	Lectura de encoders.....	152
7.3.3	<b>Lectura</b> del ADC.....	153
7.3.4	Comunicaciones. ....	155
7.3.5	Pruebas del robot. ....	156
7.3.5.1	Pruebas de error en el movimiento. ....	156
7.3.5.2	Pruebas del error en la detección de la posición de la fuente de olor. ....	158
8	Conclusiones y trabajo futuro. ....	163
8.1	Conclusiones.....	163
8.2	Trabajo futuro. ....	164

# Índice de Figuras.

Figura 2.1 Gradiente de concentración del olor. Tomado de “Olfaction and hearing based mobile robot...”[4].....	37
Figura 4.1 Placa Beaglebone tomado del manual SRM BONE de Beagleboard. ....	52
Figura 4.2 Bloques constitutivos de la Beaglebone .....	57
Figura 4.3 Sistemas Componentes de la placa Beaglebone. ....	57
Figura 4.4 Conexión Putty para una Beaglebone en Windows 7. ....	61
Figura 4.5 Entradas y Salidas de los diferentes Recursos en Beaglebone.....	63
Figura 4.6 Dimensiones de Beaglebone. ....	74
Figura 4.7 Dimensiones de los módulos XBee. ....	75
Figura 4.8 Esquemático de la fuente de alimentación de la placa adaptadora .....	76
Figura 4.9 Circuito regulador de voltaje.....	77
Figura 4.10 Motores, Rueda y Encoders Pololu. ....	78
Figura 4.11 Diagrama de distribución de pines y circuito recomendado para uso de PWM. Extraído del datasheet del fabricante Texas Instruments.....	79
Figura 4.12 Diagrama de conexiones del driver de los motores en Altium Designer. ....	80
Figura 4.13 Distribución circuital de TXB0104, obtenido del datasheet del fabricante. ....	81
Figura 4.14 Conexión de los pines del circuito buffer en Altium Designer. ....	81
Figura 4.15 Disposición y conexión del PCA9306.....	83
Figura 4.16 Footprint y conexión de XBee en la placa adaptadora. ....	85
Figura 4.17 Header de Beaglebone puerto P8 .....	86
Figura 4.18 Header de Beaglebone puerto P9 .....	86
Figura 4.19 Conexiones Ruteadas de la placa adaptadora .....	88
Figura 4.20 Vista posterior de la placa adaptadora .....	89

Figura 4.21 Vista superior de la placa adaptadora.....	89
Figura 4.22 Secciones de la placa destinadas a bloques funcionales. ....	90
Figura 4.23 placa adaptadora terminada.....	90
Figura 4.24 Proceso de montaje del chasis. ....	91
Figura 4.25 Chasis terminado.....	92
Figura 4.26 Integración electrónica y mecánica.....	92
Figura 4.27 Longitud de arco de una circunferencia.....	94
Figura 4.28 Modificación de la placa adaptadora. ....	95
Figura 4.29 Sensores SHARP y conectores en la placa adaptadora, tomado del datasheet del fabricante. ....	96
Figura 4.30 Simulador de la fuente de odorante .....	99
Figura 4.31 Proceso de montaje de la placa “nariz” .....	101
Figura 4.32 Conexiones mínimas de utilización de un Módulos XBee. Obtenida en el enlace: <a href="http://es.scribd.com/doc/58980339/Teoria-y-Programacion-Modulos-XBEE">http://es.scribd.com/doc/58980339/Teoria-y-Programacion-Modulos-XBEE</a> .....	102
Figura 4.33 Interfases seriales y USB para programación de XBee.....	102
Figura 4.34 Conexión XBee a la placa adaptadora. (plataforma personalizada) .....	103
Figura 5.1 Diagrama de flujo de la estrategia de detección del olor con una nariz electrónica.....	108
Figura 5.2 Diagrama de flujo de la detección de la fuente de luz.....	110
Figura 5.3 Diagrama de conexiones y conexión física de los detectores de luz infrarroja .....	115
Figura 5.4 Led emisores de infrarrojos utilizados. ....	116
Figura 6.1 Red Multipunto con un solo Módulo Coordinador. Tomada de la Guía de usuario XBee del enlace <a href="http://www.olimex.cl">www.olimex.cl</a> .....	124
Figura 6.2 Red Multipunto con un solo Módulo Coordinador. Tomada de la Guía de usuario XBee del enlace <a href="http://www.olimex.cl">www.olimex.cl</a> .....	124

Figura 6.3 Módulos de configuración seriales vía RS232 y USB para módems XBee.....	128
Figura 6.4 Pantalla de inicio de software X-CTU para configuración de XBee. ....	129
Figura 6.5 Reconocimiento del modem conectado al Explorer mediante X-CTU.....	130
Figura 6.6 Opción de configuración de módem XBee.....	130
Figura 6.7 Verificación del puerto serial. ....	133
Figura 6.8 Interfase grafica de la aplicación del sistema de supervisión de posición de los robots buscadores de odorante. ....	137
Figura 6.9 Código de programación gráfico de la interfase de sistema de supervisión de posición de los robots buscadores de odorante. ....	137
Figura 7.1 Niveles de integración del hardware del Robot. ....	143
Figura 7.2 Dirección del movimiento de la estrategia de búsqueda del robot.....	144
Figura 7.3 (a) cuatro agentes (b) n agentes, n celdas. ....	145
Figura 7.4 Medición de la respuesta de los encoders de las redes Pololu par diferentes relaciones de trabajo. ....	147
Figura 7.5 Conjunto de pruebas de respuestas de encoder A y B para diferentes relaciones de trabajo y frecuencia. ....	153
Figura 7.6 (a) Error introducido en la trayectoria del robot.....	157
Figura 7.7 Error permitido para detección de la fuente de luz infrarroja.....	158

## Índice de Tablas.

Tabla 4.4.1 Especificaciones del procesador ARM, del fabricante.....	51
Tabla 4.2 Configuración general del Header P8 de Beaglebone. ....	64
Tabla 4.3 Modos de operación 0-3 de los pines del header P8 de Beaglebone .....	65
Tabla 4.4 Modos de operación 4-7 de los pines del header P8 de Beaglebone .....	66
Tabla 4.5 Configuración general de los pines del header P9 de Beaglebone .....	68
Tabla 4.6 Modos de operación 0-3 de los pines del header P9 de Beaglebone .....	69
Tabla 4.7 Modos de operación 4-7 de los pines del header 9 de Beaglebone .....	70
Tabla 4.8 Lógica de funcionamiento del puente H.....	80
Tabla 4.9 Control PWM de la velocidad del motor. ....	80
Tabla 4.10 Lógica de funcionamiento de los motores .....	93
Tabla 6.1 Valor de los bits para manejo de TX y RX en ARM CORTEX, (tomado del manual de operación del fabricante) .....	134
Tabla 7.1 Apreciación del movimiento de los motores a variaciones de la señal PWM....	152
Tabla 7.2 Valores para el emisor de luz OP293B.....	154
Tabla 7.3 Valores para el emisor de luz SFH 415.....	154
Tabla 7.4 Pruebas realizadas al movimiento “recto” del robot y sus aciertos.....	157
Tabla 7.5 Distancia del robot a la fuente de luz en pruebas. ....	158



# **CAPITULO I**

## **INTRODUCCIÓN.**



# 1. Introducción.

## 1.1 Motivación.

Una de las líneas de trabajo del grupo de investigación HCTLab de la Escuela Politécnica Superior (EPS) de la Universidad Autónoma de Madrid (UAM), (Human Computer Technology Lab), es la robótica y las redes de sensores móviles inalámbricas, además de prestar soporte de hardware a otros grupos de investigación.

En éste caso particular, se produce una colaboración entre HCTLab y GNB (Grupo de Neurocomputación Biológica) para producir una aplicación de interés común, que es la implementación y adaptación de una nariz electrónica en un robot móvil, con el fin de que pueda colaborar con otros nodos de su especie, para detectar la posición de un odorante en particular y dirigir todos los nodos hacia él, a partir de allí, realizar tareas adecuadas a un amplio espectro de aplicaciones naturales, que pueden ir desde la detección y clasificación de alimentos, hasta la acción común de robot en ambientes altamente peligrosos y contaminados, pasando por tareas de detección y rescate, etc.

Esta tarea abarca varios problemas particulares intermedios, que ya han sido resueltos por HCTLab y GNB, en primer lugar la construcción del robot adecuado a la tarea; este debe ser móvil y muy dinámico, además de poseer amplia capacidad de comunicaciones inalámbricas (que no limiten su capacidad de movimiento ni las áreas de cobertura); que le permitan colaborar en la consecución de los objetivos del sistema.

Para poder cumplirla adecuadamente el robot autónomo se debe localizar en el entorno y partir de ésta localización, encontrar la fuente de olor y comunicársela a los demás para que paralelamente ellos la encuentren y se dirijan a ella, es decir una forma completamente colaborativa.

Las tareas de localización y traslado de los robots han sido logrados ya en HCTLab, así como las tareas de colaboración entre sistemas de nodos móviles, pero la integración de todos estos sistemas y sus objetivos requiere sistemas que posean mucha más versatilidad y capacidad computacional, ya que adicionalmente se debe conectar una nariz electrónica inteligente, que es un problema ya resuelto por GNB, que proporcione la capacidad de detectar la intensidad del odorante, que será la que dé origen al funcionamiento del robot.

Es decir se han cubierto varias líneas de investigación de HCTLab y GNB, a saber: Robótica, Redes de sensores inalámbricos, Diseño de sistemas basados en microprocesadores, y la línea de Neurociencia Computacional de GNB.

En la actualidad, con el desarrollo excepcional de los sistemas SoC y los sistemas embebidos de plataformas computacionales cada vez más versátiles y reducidas, se ve una amplia gama sobre las cuales se puede actuar. Es necesario para realizar la selección de la plataforma, interpretar adecuadamente los requisitos funcionales del sistema y su proyección y expansión futura, es decir su capacidad computacional, y principalmente como es en este caso particular, su capacidad de comunicaciones y la variedad de medios y protocolos estándar que dispone. En este caso se debe asegurar por lo menos tres sistemas de comunicación: la de los sistemas básicos de locomoción y localización del robot; la de la detección del olor por medio de la nariz inteligente y la de la colaboración con los demás nodos inalámbricos.

Finalmente se debe manifestar que existen varios métodos por los cuales se puede realizar la tarea colaborativa entre los nodos (robots) de la red para el caso particular que nos interesa que son:

- Que exista un agente autónomo que por sí mismo detecte y localice la fuente de odorante y cuando lo haya hecho, comunique a los demás la posición de ésta, para que se movilicen y lleguen a realizar sus tareas particulares con ella. En este caso existe una plataforma dotada de todas las características tecnológicas y otras de menor tecnología que se basan en el trabajo de la primera. Se evidencia el retardo que existirá entre la operación del único robot guía y la llegada al sitio de la fuente de olor de todos los entes del sistema.
- Que exista un robot líder que detecte y localice la fuente de olor y paralelamente y por formas indicativas (luminosas, térmicas, de colores, sistemas de transmisión, etc.) los demás entes sigan la misma trayectoria y juntos lleguen al lugar en donde se encuentra la fuente de olor. Aquí nuevamente se evidencia la existencia de una sola plataforma de mayor tecnología y equipamiento; y los demás robots acondicionados para seguir los indicativos hasta llegar al objetivo, pero se observa una mejoría en el tiempo de llegada de todos los robots al sitio de interés.
- Que todos los agentes tengan las mismas capacidades tecnológicas para que individualmente detecten la fuente de olor y al mismo tiempo colaboren con su apreciación de su dirección de movimiento con los demás robots, de forma que se reduzca al mínimo el retardo de tiempo de la búsqueda y el arribo de los agentes al sitio de interés.

## 1.2 Objetivos.

### OBJETIVOS GENERALES.

El principal objetivo de este TFM consiste en implementar una plataforma robótica, autónoma, en la que se integre una nariz electrónica inteligente, que permita al agente localizar y dirigirse hacia una fuente de olor, al mismo tiempo que colabora con el resto de agentes, para que puedan dirigirse hacia el objetivo y realizar alguna tarea establecida por un sistema jerárquico superior.

### OBJETIVOS ESPECÍFICOS.

1. Determinar el tipo de robot más adecuado para la elaboración del proyecto, ya que dependiendo de esto se determinarán metas de velocidad y eficiencia del sistema a implementar.
2. Definir el tipo de localización que deba tener el robot, ya que aunque los problemas de localización y mapeo no hayan sido resueltos en un 100% para los robots móviles a nivel global, se necesita la definición adecuada del tipo de localización que el robot va a tener, se debe tomar en cuenta el contexto de la aplicación.
3. Elegir el tipo de nariz más adecuada para producir la aplicación, ya que no se podrán montar cualquier tipo de nariz a un robot ya elegido.
4. Definir el trabajo de la red, o cómo sinónimo del mismo definir el tipo de colaboración que van a presentar los robots, si son maestro – esclavos o multimaestros los agentes de la red.
5. Definir la forma de utilización y definición de la plataforma necesaria; si es común para todos los agentes o individual dependiendo del trabajo que gestione cada uno de ellos.
6. Diseñar los acoplamientos electrónicos requeridos por la plataforma elegida, ya que de esta forma se verá mayor versatilidad y se cometerán menos errores en la consecución de los objetivos.

## 1.3 Organización de la Memoria.

El presente documento está estructurado en 8 capítulos que se resumen a continuación para proporcionar la idea general del mismo:

- **Capítulo I. Introducción.-** En este capítulo se detallan los motivos que han llevado a que se desarrolle este trabajo, además de que se revelan los objetivos que se pretenden alcanzar a la finalización del mismo.
- **Capítulo II. Estado del Arte.-** Se ha considerado que el trabajo se divide en tres grandes bloques de interés: por un lado están los sensores inteligentes o narices

electrónicas, por otro lado los sistemas colaborativos que abordan incluso los sistemas de comunicaciones y por otro lado el hardware del robot incluidos sus métodos de localización y mapeo para llevar a cabo sus tareas.

- **Capítulo III. Descripción General del Sistema.-** En esta sección se presenta un resumen de la forma en que opera el robot y sus características fundamentales, se ve a groso modo los sistemas que lo componen y su arquitectura en general.
- **Capítulo IV. Diseño e implementación de la plataforma robótica.-** Aquí se detallan las características de hardware que compone el robot, su forma y requisitos constructivos y adaptativos a las necesidades de comunicación en red, acoplamiento del sensor de olor y sistemas de sensado para posicionamiento.
- **Capítulo V. Implementación del sistema de detección de olor.-** En éste capítulo se describe el hardware y software que han sido necesarios implementar para realizar la correcta detección de los niveles adecuados de “odorante” requisito mínimo para producir su localización y la colaboración.
- **Capítulo VI. Implementación del sistema de comunicaciones y la colaboración entre agentes.-** Aquí se detallan las formas analíticas que se utilizan para programar el robot, con la finalidad de que desarrolle el trabajo colaborativo con los demás agentes y concluyan con la localización global del olor determinado.
- **Capítulo VII. Integración, Pruebas y Resultados.-** En esta sección se describe cómo se ha realizado la integración y adaptación de todo el hardware y software para producir la aplicación final, las pruebas experimentales y los resultados que ellas han arrojado, y que determinan la versatilidad y las limitaciones del sistema, así como recomendaciones que se pueden realizar para su mejora.
- **Capítulo VIII. Conclusiones y trabajo futuro.**







## **CAPITULO II**

### **ESTADO DEL ARTE**



## **2 Estado del arte.**

Tal como se mencionó en el acápite de la organización de esta memoria, el estado del arte de la investigación e implementación realizadas, se descompone en tres grandes bloques, primero se aborda la definición y aplicación de narices electrónicas, usadas como sensores inteligentes para diversidad de aplicaciones; Los robots colaborativos que no pueden prescindir de medios de comunicaciones para la consecución de sus tareas; y la ciencia de robots móviles, que ha dado de que hablar durante muchos años en entornos científicos y que actualmente se comercializan incluso para tareas de limpieza en los hogares.

### **2.1 NARICES ARTIFICIALES, GENERALIDADES Y EVOLUCIÓN.**

Estos artículos nos ayudan a comprender qué son y cuáles son los principales aspectos que se deben considerar al momento de seleccionar, diseñar e implementar un sensor de este tipo para cualquier aplicación que lo demande.

#### **2.1.1 PRINCIPIOS Y CARACTERÍSTICAS FUNDAMENTALES DE NARICES ARTIFICIALES:**

Son artículos relacionados con esta sección:

- “Electronic Nose Technology and its applications”[1].

El desarrollo de la robótica y la mecatrónica han hecho que exista un creciente interés en la biónica y la inteligencia artificial, pues se conjugan con los elementos de más alto grado que ellos pueden desarrollar; por una parte, y por otra y creo que más importante la medicina y las telecomunicaciones han creado una necesidad urgente de detección y corrección remota de enfermedades y elementos patógenos en los seres humanos, que requieren tratamientos inteligentes y que estén presentes en el sitio que convenga a cualquier hora del día o la noche.

El primer modelo de detección de gas fue introducido en 1982 por Persaud y Dodd, y desde entonces han existido experiencias muy significativas en detección de gas para aplicaciones industriales, este dispositivo se basa en la detección de muestras orgánicas y no de componentes químicos individuales dentro de una mezcla volátil. En la industria agrícola se ha logrado implementar narices artificiales para la detección de olores, aromas y calidad de alimentos, destacando por estos medios, la vida de los alimentos, la capacidad de almacenamiento, frescura, detección de residuos agrícolas, etc. Y finalmente se ha utilizado en caracterización de hongos, bacterias, perfiles de úlceras y algunas otras aplicaciones médicas.

#### **2.1.1.1 Nariz Electrónica Artificial.**

Es principalmente un instrumento inteligente que está diseñado para discriminar olores complejos utilizando una matriz de sensores, sintonizados a materiales biológicos o químicos sensibles. Constituye un elemento rápido, sencillo y poco invasivo de detección e identificación de una gama de compuestos volátiles.

Su función principal es la de imitar el comportamiento olfativo humano, y es la principal herramienta en la industria que permite caracterizar el olor de una variedad de productos de consumo. Sus primeras aproximaciones consistieron en ciertos dispositivos mecánicos y en el estudio de redox (reacciones odorantes en un electrodo), pasando por la utilización de transductores de gas selectivos que actúan sobre moléculas de gas individuales y transforman una reacción química en señal eléctrica, en la actualidad se utilizan varios tipos de material sensorial en las narices electrónicas, tales como óxidos de metales, polímeros conductores, cristales piezoeléctricos y de fibra óptica, etc.

Un estímulo olor genera un patrón característico sobre una serie de sensores, éstos se utilizan para construir una base de datos de cada olor, como una forma de reconocimiento de patrones, de esta forma se clasifican los olores en identificados o no. Típicamente una nariz electrónica consta de tres elementos: un conjunto de sensores que será expuesto a los componentes volátiles, la conversión de las señales correspondientes, y el software de análisis de los datos.

Se han identificado los siguientes principios para utilización de narices electrónicas artificiales:

#### **2.1.1.2 Biomarcadores Volátiles.**

Funcionan únicamente cuando existen condiciones favorables, en donde está presente una fuente de alimentación. Su principio de funcionamiento se basa en la característica que tienen las bacterias de generar compuestos orgánicos bajo ciertas condiciones ambientales o químicas. La intensidad de estas condiciones, está indicada por la producción de estos compuestos especiales, que en muchos casos pueden ser también la liberación de olores agradables.

Los olores se pueden utilizar para diagnosticar enfermedades y otros eventos. Recientemente se ha estudiado que la producción de compuestos volátiles puede ser empleada para identificar y diagnosticar enfermedades cancerígenas y cardiovasculares, infecciones respiratorias, diabetes, etc.

#### **2.1.1.3 Biosensores.**

Son elementos que incorporan elementos de detección biológica tales como enzimas, anticuerpos o microbios, que se asocian íntimamente con transductores físico-químicos que producen cambios en las propiedades del elemento y son detectados por los transductores y convertidos a una señal eléctrica. El transductor puede ser óptico, electroquímico, térmico o piezoeléctrico.

Estos biosensores se pueden clasificar en dos amplias categorías: sensores para detección directa y sensores con detección indirecta, los primeros se caracterizan por presentar una respuesta inmediata a la acción de la variable en el transductor con cambios específicos en él. En los indirectos, se lleva a cabo una detección preliminar con otro dispositivo que es el que genera una señal eléctrica en respuesta al transductor.

#### **2.1.1.4 Electroquímicos.**

Se basan en electrodos sensitivos que miden el cambio en la concentración de iones durante una reacción, en base a tres principios:

- Electroodos de PH de vidrio normales.
- Electroodos de PH recubiertos de membrana permeable selectiva a los gases.
- Electroodos de estado sólido.

#### **2.1.1.5 Ópticos.**

Biosensores basados en fibras ópticas y resonadores de plasma superficial, se basan en la fluorescencia o luminiscencia que producen ciertas baterías en presencia del oxígeno, estas son detectadas y transformadas en una señal eléctrica.

#### **2.1.1.6 Dispositivos piezoeléctricos basados en ondas acústicas.**

Sensibles a los cambios de masa, densidad, viscosidad y a fenómenos de acoplamiento acústico, a medida que la onda acústica se propaga a través o en la superficie del material, la velocidad o amplitud de onda cambian. Los cambios en la velocidad pueden ser controlados mediante la medición de la frecuencia del sensor, que entonces puede estar relacionada con el parámetro físico bajo consideración.

#### **2.1.1.7 Procesamiento de señales.**

También son incluidas las técnicas de procesamiento de datos, para realizar la detección la clasificación y descripción de interacción de compuestos volátiles, superficies de polímeros conductores, etc.

Actualmente se desarrollan distintos tipos de técnicas de análisis: análisis de componentes principales (PCA), análisis de conglomerados (CA), análisis de función discriminante o redes neurales de lógica difusa.

### **2.1.2 APLICACIONES DE LAS NARICES ELECTRONICAS.**

Los siguientes artículos permiten presentar la versatilidad de las aplicaciones de las narices artificiales y su proyección a futuros trabajos relacionados de investigación y aplicación.

Artículos relacionados con esta sección:

- “Applications and Advances in Electronic Nose Technology” [2].
- “Electronic Noses for Monitoring Environmental Pollution and Building Regression Model”[3]

Los aromas son mezclas complejas de compuestos volátiles presentes en el aire a una concentración que pueden ser detectados por animales a través del sentido del olfato, a estos se hace referencia como “olores” o “malos olores”, refiriéndose no a sus características mismas sino al nivel del agrado o desagrado humano.

Un aroma muchas veces se compone de un solo elemento químico, mientras que otros son verdaderos compuestos o mezclas de varios elementos o sustancias.

Una ventaja de la nariz electrónica es que puede detectar los cambios en materiales inodoros, que los seres humanos no lo pueden hacer.

Los aromas se clasifican en cuatro dimensiones cuantificables que son: Umbral, Intensidad, Calidad y Evaluación hedónica.

El valor del umbral se define por su concentración más baja; La intensidad se refiere a la fuerza percibida de la sensación de aroma y aumenta a medida de su concentración; La calidad es el uso del descriptor o palabras que describen el aroma, asociándolos con sustancias conocidas por ejemplo: terrosos, cítricos, etc. Y la evaluación hedónica constituye la agradabilidad relativa o desagrado del aroma.

La Sociedad Americana para Pruebas y Materiales (ASTM) ha clasificado 830 descriptores aromáticos. Sin embargo, las pruebas del panel de humanos han indicado que los sujetos humanos pueden identificar correctamente un promedio de menos de 100 aromas después de un buen entrenamiento de discriminación de aromas familiares. Vistos estos resultados, las narices electrónicas pueden ser construidas y calibradas para un número mucho mayor que ese, idealmente para todos estos descriptores.

Los aromas pueden ser divididos en categorías simples y complejas. Un aroma simple es un compuesto único, bien definido, en mayoría de los casos previamente sintetizados, se considera el componente principal o el director (o con mayor impacto) es en muchos casos el aroma de muchos materiales naturales. Los materiales naturales, en particular de las plantas, pueden emitir un aroma compuesto por decenas, cientos o incluso miles de especies químicas distintas.

Por otra parte, el mismo aroma puede cambiar con el tiempo si cambian las condiciones ambientales, debido a efectos sobre la volatilidad. Muchas especies de plantas también producen productos químicos muy característicos llamados aceites esenciales compuestos de mezclas simples o complejas de elementos que imparten un olor muy diagnóstico atribuible únicamente a las especies vegetales específicas o taxones.

Las diferentes volatilidades de las especies moleculares que componen los aromas florales se consideran importantes en el desarrollo de productos dentro de la industria cosmética y perfumería. Los compuestos más volátiles representan las "notas altas", que producen un impacto olfativo inmediato, mientras que las "notas bajas" son aromas más persistentes y sutiles por lo general menos volátiles. Estos dos componentes caracterizan la estructura fundamental de aroma de un perfume o colonia con un olor particular.

Los hidrocarburos por lo general no presentan olores de interés o de un carácter bien definido, aunque ciertos hidrocarburos insaturados tales como alquenos cíclicos han sido identificados y asociados con las notas típicas y agradables, como afrutado, verde, y olores florales.

Los umbrales de detección de los aromas pueden considerarse como gases en condiciones de temperatura y presión específicos generalmente altos, y los bajos umbrales para bajos niveles de presión.

La gran especificación en la industria y sus continuos avances en los sistemas de control electrónico que utilizan, ha hecho que en los últimos veinte años, haya habido un enorme impulso en la tecnología para el desarrollo de dispositivos de sensado.

Por supuesto que las narices electrónicas no han sido la excepción y sus avances han devenido en relación con el diseño de sensores, mejoras en los materiales, innovaciones en el software y en los avances en el diseño de microcircuitos y sistemas reconfigurables, todo esto sobre la base de nuevos principios y mecanismos de detección, y además estrechamente relacionados con la aparición de nuevas aplicaciones.

Un sistema de nariz electrónica consiste típicamente en una matriz de sensores múltiples, una unidad de procesamiento de información tal como una red neuronal artificial (ANN),

el software con algoritmos de reconocimiento de patrones digitales, y referencia de la biblioteca de bases de datos. La matriz de sensores de reacción cruzada se compone de diferentes sensores incrementalmente elegidos para responder a una amplia gama de clases químicas y discriminar diversas mezclas de posibles análisis.

El artículo referenciado vuelve a repetir la historia de la aparición de la tecnología de sensado tipo olfativa, que tuvo orígenes en 1982 con multisensores, y que ha tenido considerables avances en el campo de los aromas, la microelectrónica, la bioquímica y la inteligencia artificial que hacen posible el desarrollo de dispositivos capaces de medir y caracterizar aromas volátiles liberados de una multitud de fuentes. Estos equipos se han diseñado para obtener mediciones repetibles, lo que permite la identificación y la clasificación de aromas.

A diferencia de otros instrumentos de análisis, estos dispositivos permiten la identificación de mezclas de muestras orgánicas como un todo sin tener que identificar las especies químicas individuales que la componen. Se han desarrollado diferentes prototipos de dispositivos artificiales para discriminar complejas muestras que contienen diferentes tipos de compuestos orgánicos volátiles. Estos dispositivos utilizan diferentes tipos de sensores incluyendo semiconductores CMOS; polímeros conductores electroactivos, sensores ópticos, ondas acústicas de superficie y sensores de gas electroquímicos.

#### **2.1.2.1 Desarrollo conceptual de la nariz electrónica.**

Los primeros estudios de medición de aromas se centraron en la medición de energía eléctrica de un fino rocío de agua, pues se podía detectar pequeñas cantidades de compuestos aromáticos, se desarrolló un microelectrodo para medir el flujo de corriente producida por la intensidad de los aromas, y sus sustancias químicas componentes, de esta forma revestir de distintos componentes dicho electrodo, y se daría como resultado la medición de distintos aromas.

A partir de estos experimentos se establece que la construcción de matrices con diversos electrodos sensibles a niveles diferentes de aromas, podían ser utilizadas para una mejor respuesta y discriminación de un gran número de diversos aromas.

El término "nariz electrónica" es definido como "un instrumento que comprende un conjunto de sensores químicos electrónicos con especificidad parcial y sistema de reconocimiento de patrón apropiado, capaz de reconocer olores simples o complejos"



### **2.1.2.2 Instrumentación de una Nariz Electrónica.**

Para la implementación de un sistema de detección de olores es necesaria la implementación de los siguientes componentes:

- Sistema de suministro del aroma, que es el que reparte las moléculas volátiles de la fuente hedónica hacia el arreglo de sensores para su detección.
- Cámara de ubicación de los sensores, que debe mantener una temperatura y humedad específicas.
- El transductor, que convierte las señales químicas en eléctricas y la amplifica para su posterior tratamiento.
- Un convertidor A/D, que permite convertir esa señal eléctrica proporcional en palabras digitales para su tratamiento en elementos discretos.
- Un controlador, elemento inteligente basado en microprocesadores, que realiza el análisis estadístico de la muestra y la enmarca en su tipo determinado.

Se establece que la parte más sensible del sistema es el apartado de suministro del aroma, pues es el elemento que introduce el aroma a la cámara del sensor haciendo que reciba con todas sus características el olor, y será detectado y convertido en la cámara para su posterior procesamiento.

### **2.1.2.3 Análisis de la información de las narices electrónicas.**

Las salidas de los sensores deben ser interpretados para que puedan ser útiles a las aplicaciones determinadas, éste análisis puede ser hecho de tres formas:

- Mediante gráficos de diversos tipos.
- Mediante análisis de las componentes principales, reduciendo los datos e identificando las causas de la variabilidad.
- Mediante redes neuronales

La elección del sistema depende del tipo de datos de entrada disponibles adquiridos de los sensores y el tipo de información que se busca. La forma más simple de reducción de datos es el análisis gráfico útil para comparar las muestras o los elementos de identificación del aroma.

Los artículos analizados reflejan el amplio campo de uso y aplicaciones para las narices electrónicas. Se referencian ejemplos en el ámbito de la medición de la frescura, la calidad, la madurez y la vida útil de alimentos perecederos, vegetales, carnes o pescados, así como en el ámbito de los productos lácteos o del vino. También se referencia el uso de narices electrónicas en el fitopatología, identificación de plantas o en patologías médicas. Por último resaltar el uso de las narices electrónicas para clasificar y cuantificar diversos

aromas y gases en el ámbito de la contaminación ambiental. Esta aplicación ayuda a resolver problemas graves porque están directamente relacionados con las enfermedades crónicas. Existen muchos factores que causan la contaminación del aire, unas naturales como las condiciones climáticas, la humedad, la topografía, pero también las derivadas de la actividad del hombre como son emisiones de derivados del carbón, de los vehículos, fábricas, etc. A pesar de la existencia de normas de protección, el control del cumplimiento de ellas es costoso y muchas veces muy técnico y complejo.

Este ámbito es uno de los nichos sobre los cuales es ideal emplear narices electrónicas, por su rápida respuesta y facilidad de operación, en donde dos aspectos intrínsecos se deben considerar: la selectividad y la sensibilidad, que son definidas como la capacidad del sensor para detectar el parámetro de interés y los niveles mínimos que puede medir, respectivamente. Para realizar estas mediciones se utilizan narices basadas en sensores MOSFET de óxido metálico. El valor de la concentración de oxígeno en la superficie del sensor modifica la concentración de portadores en el dispositivo semiconductor, aumentando la región de vaciamiento y en consecuencia modificando las características eléctricas del dispositivo. De esa forma la nariz actúa como sensor de oxígeno, que conjuntamente con otras características sirven para detectar gases ambientales como CO y CO<sub>2</sub>.

## **2.2 APLICACIONES DE NARICES ELECTRONICAS MONTADAS SOBRE ROBOTS.**

En estos artículos se encontrarán algunas definiciones útiles al momento de montar una nariz artificial en robots con fines específicos.

Artículos relacionados con este tema:

- “Olfaction and hearing based mobile robot navigation for odor/sound search”[4].
- “Olfaction based mobile robot navigation”[5].

Las tecnologías biónicas establecen que los procedimientos principales de sus construcciones deben imitar perfectamente los sentidos biológicos, en ese contexto es necesario que se determinen las características necesarias para producir este objetivo en robots que utilizan narices electrónicas para detección de aromas específicos en aplicaciones específicas.

En este artículo se presentan las características de movimiento de un robot que localiza aromas y sonidos, y los algoritmos específicos de marcha del robot para lograr esos objetivos de forma automática, de tal forma que puede modificar su posición y velocidad.

Al mismo tiempo el robot puede comunicarse con otros robots a través de una red de sensores inalámbricos.

### **2.2.1 Tecnologías de detección y movimiento de robots.**

Se han utilizado tecnologías para la detección utilizando sistemas MOS, sensores de gas de metal, algoritmos biométricos, microbalanzas de cristal de cuarzo, de gas, robots en espacios tridimensionales, robots que emulan movimientos de langostas, lombrices y redes neuronales para predecir la distancia, conjugados con sensores de visión, flujo de aire, temperatura y humedad, algoritmos de evitación de obstáculos, etc. Además, las búsquedas de fuentes de olor se pueden lograr mediante la cooperación de multi-robots en lugar de los robots individuales.

En este artículo se describe un sistema de multi-robot usando el olfato, el oído y el tacto para rastrear el destino de olor/sonido este sistema se compone de un robot de olfato y dos robots de audición. Los tres robots comprenden una red inalámbrica y cada robot es un nodo de la red. En primer lugar, el robot de olfato realiza un seguimiento y búsquedas para la fuente de olor. Una vez que se encontró la fuente de olor, sonará y enviar señales para llamar a los otros dos robots auditivos. Posteriormente, los dos robots de audición buscan el sonido para encontrar a su compañero.

El robot puede ajustar su dirección de la marcha de acuerdo con la desviación entre la dirección de la marcha actual y la dirección de la marcha esperada, utiliza un control tipo PID para el ajuste normal del error de su posición. La eficacia de este método es verificado por experimentos. Los resultados muestran que los robots olfato/audición pueden buscar fuente de olor/sonido con eficacia y eficiencia.

### **2.2.2 Estructura del robot de olfato.**

Busca la fuente de olor, una vez encontrada suena para llamar a los otros robots de audición, para que puedan acercarse a él. Para realizar esta tarea está equipado con dos sensores de gas, sensores de flujo de aire, sensores de temperatura, interruptores de contacto, sensor magnetorresistivo y un zumbador.

Existen en él tres sensores de gas separados 120 ° entre ellos, con la finalidad de ajustar su dirección mediante la comparación de las salidas de los tres sensores. Dos sensores de flujo de aire, para recibir la velocidad del viento y equilibrarla según las recibe en su lado izquierdo y derecho. También existe un circuito de compensación de temperatura que utiliza RTD Pt1000 y un CI LM35 para medir la temperatura del ambiente, dos switch de contacto que permiten percibir si el robot choca o no con la fuente de olor y un sensor

magnetorresistivo para detectar el ángulo de partida del robot, mediante la medición de la intensidad de campo de los ejes horizontal y vertical.

Esta montado con un DSP de 32 bits a 150 MHz que integra 12 canales de E/S y un conversor A/D y modulación PWM. En donde el controlador (DSP) adquiere todas las salidas de los sensores anteriormente mencionados y realiza el procesamiento de las señales, ejecuta un algoritmo de búsqueda, acciona un motor y ajusta la velocidad y el curso, además de que se comunica con la unidad Wireless, a través de una interfase serie.

Todo este sistema se conecta a un PC de tal forma que se indica el proceso de seguimiento de la señal de olor, en comparación con sus posiciones relativas actuales, la información del sensor de gas, la concentración y dirección del viento y la temperatura. Emite también su posición actual con la finalidad de que los robots sonoros puedan identificar su posición y realizar las tareas de localización del sonido cuando este se emita.

Un módulo ZigBee se utiliza como transceptor para la comunicación inalámbrica que funciona a 2,4 GHz.

El robot es un vehículo de tres ruedas, dos motrices y una rueda de dirección, conectados a las dos ruedas motrices se encuentran dos motores. La dirección de giro de éstos es determinada por la dirección de la corriente y la rotación por el ciclo de trabajo del PWM; para reducir las interferencias electromagnéticas que introducen los motores se han utilizado opto-acopladores en las señales de control de los motores.

### 2.2.3 Distribución del campo de olor.

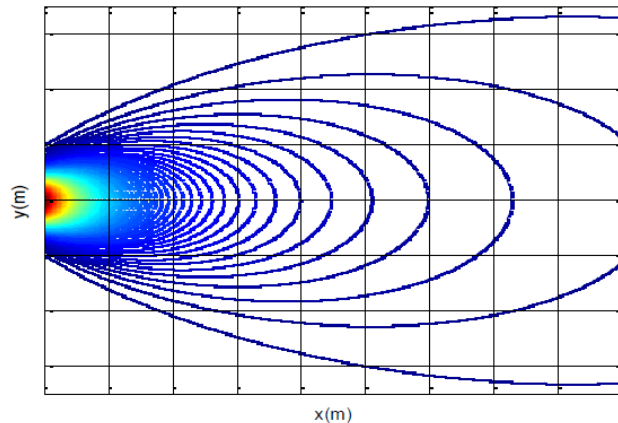
Como la velocidad de difusión de la molécula de olor es siempre más lenta que la velocidad del viento, la forma de transporte se determina por advección de aire (variación de un escalar en un punto dado por efecto de un campo vectorial), así la ecuación de distribución de la concentración se puede establecer utilizando el modelo de difusión gaussiana, si se ve afectada por turbulencia, puede ser estimada utilizando la ecuación de difusión turbulenta, o cuando la velocidad del viento es nula solamente será definida por la velocidad de difusión de la molécula. En este trabajo utiliza condiciones de brisa estable, y usa la ecuación de distribución de la concentración de gas sobre la base del modelo integral de difusión móvil. EC. (2.1) (tomada de [4]).

$$C(x, y, z, H) = \int_0^{+\infty} \frac{Q}{(2\pi)^{\frac{3}{2}} \sigma_x \sigma_y \sigma_z} \cdot \exp\left[-\frac{(x-ut)^2}{2\sigma_x^2}\right] \cdot \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \cdot \left\{ \exp\left[-\frac{(z-H)^2}{2\sigma_z^2}\right] + \exp\left[-\frac{(z+H)^2}{2\sigma_z^2}\right] \right\} dt$$

**EC (2.1)**

Donde  $C$  es la concentración de gas,  $Q$  es la intensidad de la fuente de gas  $u$  es la velocidad media del viento,  $x$  es la distancia a favor del viento,  $y$  es la distancia de viento cruzado,  $z$  es la distancia de la fuente de gas desde el suelo,  $H$  es la altura de la fuente de gas,  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  son las desviaciones estándar de la concentración de gas  $t$  es el tiempo de movimiento.

Produciéndose un gradiente de concentración del aroma, de la forma que indica la **¡Error!  
No se encuentra el origen de la referencia.**



**Figura 2.1 Gradiente de concentración del olor. Tomado de “Olfaction and hearing based mobile robot...”[4]**

En donde se observa la forma irregular y la distribución de la concentración de campo de olor.

## **2.2.4 Rastreo de la fuente de olor.**

Para producir el rastreo de la fuente de olor se tiene que determinar las condiciones de dirección y velocidad del viento, ambiente, temperatura; y se introduce el principio del control de la partida del movimiento del robot basada en la dirección. El robot de olfato utiliza un sensor magnetorresistivo, controlador PID y PWM de mapeo para ejecutar un control de bucle cerrado para su velocidad y dirección. Finalmente se utiliza un algoritmo de búsqueda paso a paso de la siguiente forma:

### **2.2.4.1 Estrategia de búsqueda.**

El robot encuentra la fuente de olor según el gradiente de concentración en un punto determinado, la distribución del olor está determinada por el viento existiendo un pico muy estrecho. El robot sigue este pico haciendo un zigzag por el borde de la curva, cuando la velocidad del viento es baja el robot busca la fuente de olor a lo largo de la gradiente de concentración de forma adaptativa, para velocidades del viento relativamente pequeñas,

se mezclan las estrategias, y se deben controlar con cuidado la dirección y movimiento de los motores, así como los umbrales de calibración de los sensores, dividiendo la estrategia de movimiento en varias fases

#### **2.2.4.2 Control de movimiento basado en la partida.**

Cuando el robot se mueve muy rápido, la lenta respuesta de los sensores hace que exista un error constante en el éxito de la búsqueda, por eso que se debe ralentizar el robot para mejorar el éxito de la búsqueda, El robot debe automáticamente y de forma estable ajustar la velocidad y dirección de acuerdo a su desviación de la fuente de olor y la marcha actual de la siguiente forma:

El ángulo de partida se calcula a través del polo norte magnético, ayudado por el chip magnético que mide este campo en los dos ejes (X y Y), montando dos sensores magnetorresistivos mutuamente perpendiculares.

El robot determina su próximo ángulo de giro dependiendo de la dirección del viento y la concentración del gas y se suman los ángulos de partida y el nuevo ángulo obtenido, cambiando la velocidad continuamente con el uso del algoritmo PID. El error de salida del controlador PID, especifica las características de las constantes Kp, Ki, y Kd, para su mejor funcionamiento.

#### **2.2.5 Navegación en robots móviles basados en olfato.**

Basado en el principio de que los insectos utilizan sus sentidos olfativos para muchas actividades, se aplican los conceptos para el manejo de un robot móvil, en este artículo se analiza cómo un robot extrae esta información olfativa para su desplazamiento. Compara tres algoritmos de navegación en un ambiente con obstáculos, basados en tres estrategias: la quimiotaxis, el algoritmo de la polilla y la estimación de la geometría del olor, aplicados a un robot Super Scout II.

El movimiento de robots autónomos está basado en usar máquinas inteligentes conectadas a una infinidad de sensores que les permitan “observar” los trayectos que caminan. El primer método que utilizan estos robots para moverse se basa en la quimiotaxis del movimiento de las bacterias, es decir organizan un movimiento partiendo de condiciones iniciales, si las condiciones mejoran, continúan con esa dirección, si las condiciones empeoran, dirigen su movimiento en otro sentido; de igual forma en los robots implementados con narices electrónicas, se presenta un movimiento inicial, si las condiciones de búsqueda del odorante mejoran, mantiene la dirección del movimiento, sino lo cambian hacia una nueva. La aplicación pretende disminuir los eventos aleatorios de los errores instantáneos.

En cambio la estrategia de los gusanos de seda es una vez detectado el odorante, inician un movimiento de zigzag en torno a él observando la mejora de dirección y encontrando la fuente del odorante, si pierde el rastro del mismo, gira en círculo nuevamente hasta volver a encontrarlo y nuevamente hacer los movimientos de zigzag. Finalmente como en el caso anterior es el método de encontrar un borde de la concentración y seguir su gradiente de concentración.

### **2.2.6 Arquitectura de control del robot.**

Está basada en el comportamiento de los actuadores acoplados en capas (*subsumption*) a través de procesos independientes que implementan en conjunto todas las metas del movimiento del robot.

El sistema de sensado es bastante parecido al sistema del robot anterior, utiliza una matriz de sensado de cuatro sensores de óxido metálico, el uso conjunto de los sensores disparan el algoritmo de movimiento. La resistencia del sensor que actúa como elemento principal, se mide a través de una tarjeta controladora de 16 canales de entrada y devuelven las señales a unos controladores de flujo másico, internamente se utiliza una capa de red neuronal para identificar las diferentes concentraciones del gas, ajustándose los pesos de la red en forma continua para obtener la dirección de la concentración hacia donde debe dirigirse el robot.

El modelado de los olores consiste en determinar, como en el caso anterior, la pluma o borde del olor del aroma buscado. Las plumas son modeladas por un conjunto de ecuaciones diferenciales, este enfoque es complicado, porque se debe determinar la geometría del ambiente.





## **CAPITULO III**

### **DESCRIPCIÓN GENERAL DEL SISTEMA.**



### 3 Descripción General del Sistema.

Tomando en consideración de lo mencionado en la sección 1.1 en donde se especifica que la tarea colaborativa podía realizarse de varias formas, se ha considerado, que la forma más eficiente, que ahorra recursos, físicos, económicos y computacionales, es hacer todos los robots con igual arquitectura y la misma plataforma, de manera que estos tengan las mismas posibilidades de culminar su trabajo con éxito, y de esta forma hacer de la tarea colaborativa se realice en el menor tiempo posible, incluyendo a los agentes que puedan contener errores en su búsqueda.

Se detallan a continuación las razones por las que se ha elegido dicha alternativa:

- Para el primer caso es decir un agente completamente equipado, y los demás agentes, como entes pasivos (de la detección de olor únicamente) la desventaja fundamental es el tiempo en el cual todos los agentes llegarían a la fuente de olor, pues el agente central (totalmente equipado), llegará a la fuente de odorante y comunicará su localización, dependerá entonces de cada uno de los demás agentes la rapidez en la que encuentren las coordenadas requeridas. Incluyendo el error del robot principal y los errores en que pueden incurrir todos los robots en sus desplazamientos. La ventaja está solamente en la implementación de la nariz en un solo elemento.
- La segunda alternativa, aunque mejora el tiempo de búsqueda, porque permanentemente los demás agentes están buscando y persiguiendo al robot principal, tampoco es tan rápida porque los robots incluyen error cuando buscan al robot guía, y este incluye también error en la búsqueda. Su ventaja nuevamente se presenta por el hecho de utilizar un solo elemento con la nariz artificial, pero ya incluye las desventajas de búsqueda del robot guía por sí mismos, de todos los demás robots.
- Finalmente llegamos a la tercera que es la elegida, se reduce el tiempo de búsqueda ya que todos los robots están equipados con narices, y además corrigen errores con la tarea colaborativa, aunque es la más cara económicamente porque a todos se les debe proporcionar las mismas características de hardware, el costo computacional es menor ya que tienen la misma plataforma de software, al contrario de las demás que se necesitaba una para el robot guía, y otra para sus colaboradores.

#### 3.1 Descripción general del robot.

La arquitectura del robot construido está basada en la desarrollada en [6], es un robot móvil que se desplaza hacia adelante, atrás, gira y se dirige solamente por la tracción de

sus dos únicas ruedas motrices, ya que posee dos ruedas más que solamente están para darle estabilidad al robot.

Este robot es capaz de medir su desplazamiento y encontrar la orientación de sus movimientos en cada trayectoria, pero se localiza únicamente con su referencia original, que para él sería el punto (0,0) de coordenadas, la razón de esto es que la característica fundamental del robot es poder producir la colaboración entre todos los demás agentes al encontrar entre todos la fuente de odorizante, es decir el robot no puede enviar su localización exacta a los demás robots, sino la dirección hacia donde se está moviendo en esa búsqueda, de ahí que la independencia de cada robot es importante para producir la colaboración, sino se llegaría al punto en estar nuevamente cayendo en una de las dos alternativas anteriores, sino que con un sistema de localización y posicionamiento de un solo robot a través de los demás.

La primera tarea que debe realizar el robot, es su movimiento y conseguir su orientación, esto lo realiza con un conjunto de chasis, ruedas, motores y encoders que dan y recibe la información necesaria desde y hacia el control inteligente que puede ser desarrollado por varias plataformas, de las cuales se debe seleccionar la más eficiente.

A continuación se debe realizar la función de oler el compuesto, para ello, el robot se comunica digitalmente con la nariz artificial, determinando a cada instante la intensidad del olor e investigando el “rumbo” de su procedencia, de ahí que tomará dicho rumbo y seguirá monitoreando la intensidad del olor. Esta nariz artificial se puede comunicar con el robot de varias formas, desde poseer su caracterización como una señal analógica equivalente, hasta digitalizar sus componentes y colocarse en uno de los varios protocolos de comunicaciones existentes. En el caso particular que manejamos, la nariz se comunica a través del protocolo I<sup>2</sup>C, pues es la forma en que el grupo de investigación GNB ha considerado debe darles operatividad a sus componentes.

Así mismo cuando el robot haya determinado la dirección de su movimiento, la comunicará a sus compañeros. Para ello, también existen diferentes alternativas, por la pequeña área de cobertura en la que tendrán que desenvolverse los agentes, podría ser desde radiofrecuencia en señales analógicas hasta comunicaciones digitales inalámbricas como bluetooth, Zigbee, Wireless LAN, etc. En el caso particular nuestro, para esto se ha dispuesto de un módulo de comunicaciones Zigbee, que realiza la transferencia de los datos entre los agentes, se lo ha elegido por sus funcionalidades de operación y programación especialmente. Estos módulos están configurados en malla para producir el efecto colaborativo entre los n nodos que pueda tener la red, es decir el sistema.

Para todo ello, los sistemas de diferentes características y fabricantes, se deben acoplar entre ellos y con la tarjeta madre de control, especialmente en niveles de voltaje y corriente; este procedimiento se realiza en una tarjeta diseñada y construida para el efecto en el HCTLab.



## **CAPITULO IV**

### **DISEÑO E IMPLEMENTACIÓN DE LA PLATAFORMA ROBÓTICA**





## 4 Diseño e implementación de la Plataforma Robótica.

Para entender la forma en la que se ha desarrollado el robot, de acuerdo a sus necesidades, se debe identificar la labor principal que debe desarrollar y de esta forma seleccionar y diseñar los elementos más adecuados para su operación.

En primer lugar el robot necesita un control inteligente, que tenga una alta capacidad de procesamiento y sobre todo posea como requisito mínimo los elementos de comunicaciones básicos que necesita, pero a más de ello, algunos más para darle desarrollo futuro sostenible, que pueda programarse de forma intuitiva y al 100% de sus capacidades.

Existen varias alternativas de plataforma embebida, que pasan desde las tarjetas Arduino con poca, mediana y gran capacidad computacional, pasando por tarjetas más eficientes como las Raspberry PI, IGEP, VAR-SOM-AM33, Beaglebone, Beagleboard, etc.

El sistema elegido es una placa Beaglebone original (blanca), que posee un procesador ARM Cortex-M3, las razones son diversas. Partiendo del hecho mencionado anteriormente que las dos principales características que debe poseer la plataforma son: Una gran capacidad computacional y la posibilidad de presentar varias formas de comunicación con el exterior, la placa Beaglebone posee ampliamente dichas características, además de que presenta una tercera: bajo costo, pues esta tarjeta se la puede adquirir a un precio de € 78. Es decir presenta las características ideales para la implementación de la plataforma.

Esta placa, como se va a describir más adelante, posee niveles de entrada y salida de voltaje de 5Vdc, 3.3Vdc y 1.8 Vdc, lo que hace que no todos los elementos que se van a conectar a ella, sean “compatibles” en niveles de tensión, por lo cual se vio la necesidad de diseñar y construir una placa adicional, que además de adaptar todos estos niveles de tensión, también produzca el óptimo manejo de potencia que necesitan los motores, es decir un driver que haga que, además se protejan a los puertos de la placa.

Para la comunicación de la Nariz con la Beaglebone, originalmente se pensó hacerla de dos métodos bastante adecuados, que eran la estrada analógica directa, ya que la placa posee 8 entradas analógicas provistas de un conversor analógico – digital que producía para cada nivel una sencilla palabra digital de 12 bits, o hacerlo de forma digital, utilizando el protocolo CAN bus, ya que la placa Beaglebone posee dos entradas y dos salida para este protocolo, que además puede derivarse a un sistema de supervisión mediante protocolos de redes industriales. Estas alternativas fueron descartadas, no por ser

ineficientes, sino porque la arquitectura de la nariz diseñada en el grupo GNB, utilizaba comunicación I<sup>2</sup>C, pero afortunadamente, no se debió hacer ningún acondicionamiento de esta señal, ya que la placa Beaglebone tiene dos canales de entrada y salida para puerto I<sup>2</sup>C.

Por otro lado para la comunicación entre los entes de la red, se decidió primeramente tomar en cuenta las posibilidades de manejo de la placa; la condición general, es que deban ser sistemas inalámbricos pues el robot es completamente autónomo, entonces podía ser wifi, RF, etc. El sistema elegido es la red ZigBee, pues la placa Beaglebone maneja directamente cinco puertos seriales UART, y como el sistema ZigBee se alimenta en las líneas de transmisión y recepción de forma bastante intuitiva de señales seriales, se utiliza este tipo de plataforma, y especialmente porque son de fácil programación cada uno de los módulos y no necesitan hardware adicional, de hecho los niveles de voltaje que manejan son los mismos que maneja la tarjeta Beaglebone, 3.3Vdc.

## **4.1 Placa Beaglebone.**

Beaglebone es una placa del tamaño de una tarjeta de crédito construida por Texas Instruments, que tiene la funcionalidad de un ordenador en esas dimensiones. Posee un sistema operativo Linux Embebido, aunque también puede funcionar con Android como sistema operativo; con el cual se le puede dar precisamente la funcionalidad de un ordenador. Posee un procesador ARM Cortex-A8 AM335x de 720 MHz, con una completa funcionalidad de puertas de entrada salida completamente configurables y de aplicación general o particular a cada una de sus “capas” de expansión diseñadas para ella. Sus características fundamentales son:

- Memoria RAM DDR2 de 256 MB.
- Acelerador gráfico 3D.
- Procesador de serie ARM Cortex-M3.
- Doble procesador RISC.
- Conectividad USB, Ethernet y dos header de 46 pines, para darle funcionalidad completa.
- Memoria SD card de 4GB con un sistema Angstrom Distribution incorporado.

Como se observa en la explicación obtenida en la página de ARM [12]; El procesador ARM Cortex™-M3 es el procesador de 32 bits líder en la industria para aplicaciones en tiempo real altamente deterministas, desarrollado específicamente para permitir a los desarrolladores construir plataformas de bajo costo y alto rendimiento para una amplia gama de dispositivos, incluyendo microcontroladores, sistemas de carrocería de automóviles, sistemas de control industrial y redes inalámbricas y sensores, y muchas

otras más. El procesador ofrece un excelente rendimiento computacional y la respuesta del sistema es excepcional a las necesidades, mientras que satisfacen las restricciones de potencia. El procesador es altamente configurable y permite una amplia gama de implementaciones.

A continuación la Tabla 4.4.1 muestra las especificaciones proporcionadas por el fabricante:

**Tabla 4.4.1 Especificaciones del procesador ARM, del fabricante**

#### ARM Cortex-M3 Especificaciones

ARM Cortex-M3 Características	
Soporte ISA	<a href="#">Thumb®</a> / <a href="#">Thumb-2</a>
Tubería	3-etapa
Eficiencia Rendimiento	3.32 CoreMark / MHz *
Eficiencia Rendimiento	1,25 / 1,50 / 1,89 / DMIPS MHz **
Protección de la memoria	Opcional 8 MPU región con subregiones y la región fondo
Interrupciones	Interrupción no enmascarable (NMI) + 1 a 240 interrupciones físicas
Interrumpir los niveles de prioridad	8 y 256 niveles de prioridad
Wake-up controlador de interrupciones	Hasta 240 alarmas de despertador
Modos de espera	Integrado WFI e instrucciones WFE y dormir sobre la capacidad de la salida. Sleep & Señales sueño profundo. Modo de retención opcional con el kit de administración de energía ARM
Manipulación de bits	Instrucciones Integrados Bit Anillamiento
Instrucciones mejoradas	Divide Hardware (2-12 ciclos), de un solo ciclo (32x32) Multiplicar, Apoyo en Matemáticas saturada.
Depurar	Opcional JTAG y <a href="#">depuración de serie-Wire</a> Puertos. Hasta 8 puntos de interrupción y 4 puntos de observación.
Rastrear	Opcional Instrucción <a href="#">traza (ETM)</a> , seguimiento de datos (DWT), y la instrumentación de seguimiento (ITM)

Este procesador está embebido en la placa Beaglebone, que como decía en la teoría anteriormente expuesta, posee completa funcionalidad de comunicaciones, a través de 92 pines de sus receptáculos, a más de las entradas de USB y Ethernet, que sirven de comunicación básica.

## 4.2 Beaglebone características Generales.

En este suplemento se describen las características del sistema integrado Beaglebone de bajo costo.

Tal como reza textualmente el manual técnico de la placa: “Beaglebone System Reference Manual” [7].

Las siguientes son las especificaciones principales existentes en la tarjeta Beaglebone; en la Figura 2.1, se puede observar la placa con sus dispositivos generales:

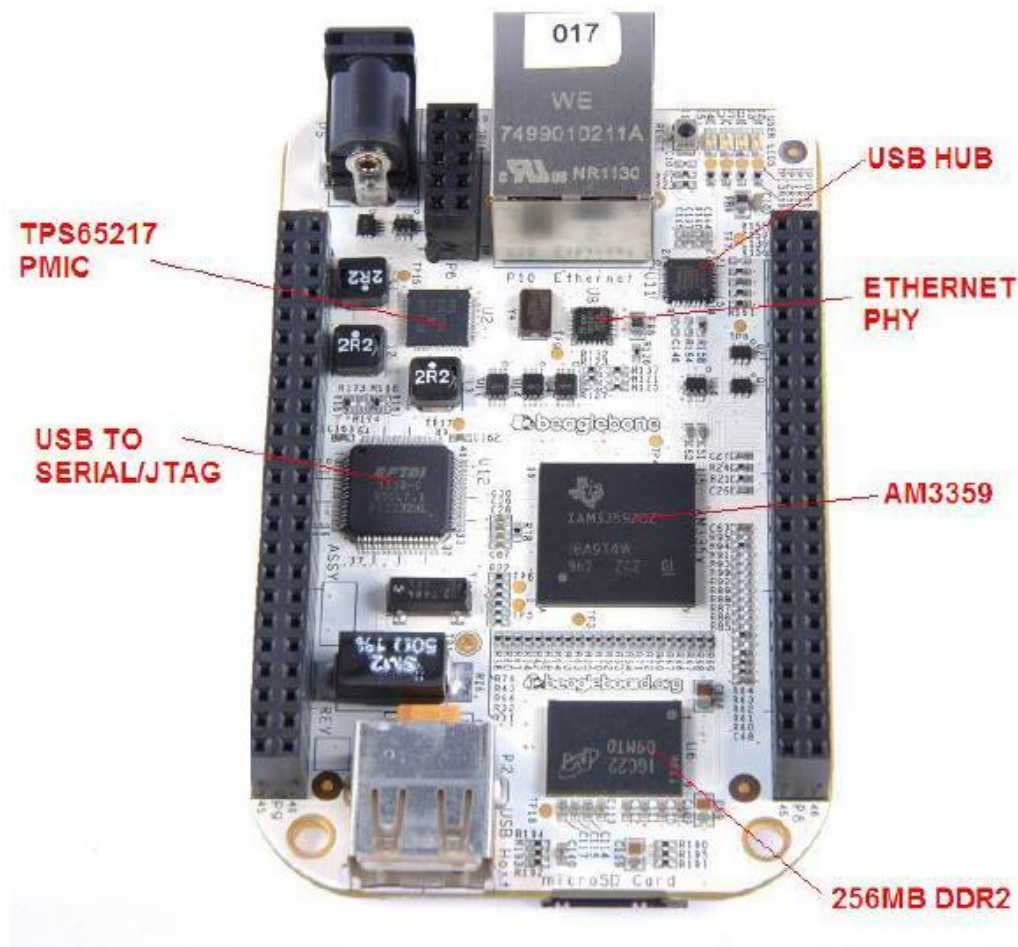


Figura 4.1 Placa Beaglebone tomado del manual SRM BONE de Beagleboard.

### 4.2.1 Procesador:

Utiliza un procesador AM3359 AM3358. La velocidad real del procesador será determinada por los dispositivos reales suministrados, y la forma de conexión.

### **4.2.2 Memoria.**

Memoria DDR2 de 16 bits, puede configurarse como de 128MB o 256MB, la configuración estándar es 256 MB a 400 MHz. Contiene una memoria EPROM de 32 KB, con la información básica de la tarjeta.

### **4.2.3 Interfase USB**

Concentra dos puertos USB, como funcionalidad para operar con:

- Comunicación serial USB.
- JTAG por USB.
- Puerto de acceso USB al procesador.

Cuando se conecta a una PC cada una de ellas se mostrará como un puerto en la PC.

### **4.2.4 Puerto serie.**

Posee un puerto serie a través de un UART0 con doble canal FT2232H USB, incluyen señales de Tx, Rx, RTS, CTS.

Una sola EEPROM se proporciona en el FT2232H para permitir la programación de la información del proveedor de manera que cuando se conecta, la tarjeta puede ser identificada y el controlador apropiado instalado.

### **4.2.5 Puerto JTAG.**

El segundo puerto en el FT2232H será utilizado para el puerto JTAG. Conexión directa con el procesador se realiza desde el FT2232H. Hay un encabezado JTAG proporcionado en la placa como una opción adicional.

### **4.2.6 Puerto USB0.**

El HUB se conecta directamente al puerto USB0 en el procesador. Esto permite que el puerto sea accesible desde el mismo conector USB como los puertos serie y JTAG.

### **4.2.7 Conector Micro SD.**

La tarjeta está equipada con un conector microSD para actuar como fuente de arranque principal. Una tarjeta microSD de 4 GB se suministra con cada placa. En el conector se admiten tarjetas SD de mayor capacidad.

#### **4.2.8 Puerto USB1.**

En el tablero hay un solo conector USB tipo A con el apoyo total Host LS / FS / HS que se conecta a USB1 en el procesador. El puerto puede proporcionar alimentación de encendido / apagado y hasta 500mA de corriente a 5V. Bajo la alimentación USB, la tarjeta no será capaz de suministrar el 500 mA completamente, pero debe ser suficiente para suministrar suficiente corriente para un dispositivo USB de baja potencia.

Puede utilizar una configuración de teclado / ratón inalámbrico o puede agregar un HUB para el teclado y el ratón de interfaz estándar si es necesario.

#### **4.2.9 Puerto USB Cliente.**

Acceso a USB0 se proporciona a través del concentrador USB integrado. Se mostrará en un PC como un dispositivo USB estándar.

#### **4.2.10 Fuentes de poder.**

El tablero puede ser alimentado por un puerto USB de un PC o desde una fuente de alimentación de 5 VDC opcional. La fuente de alimentación no se suministra con la tarjeta y debe ser una fuente de alimentación conectada a tierra. El cable USB se incluye con la tarjeta.

Cuando se alimenta desde USB, la tarjeta está limitada a 500 MHz. El consumo de energía FT2232H no puede proporcionar la corriente de 500mA para el proceso de arranque. Para la operación 720 MHz, se requiere alimentación de CC.

Se puede suministrar energía a través de un conector de 2,1 mm x 5,5 mm cuando se conecta a una fuente de alimentación positiva con valor nominal a 5 Vcc  $\pm 0,1$  V y 1A.

#### **4.2.11 Botón Reset.**

Cuando se presiona y se suelta, provoca un reset de la placa. Debido al pequeño tamaño del interruptor, no se experimenta una gran cantidad de rebotes al pulsar el interruptor.

#### **4.2.12 Indicadores de estado.**

Hay cinco LED's verdes en el tablero. Cuatro pueden ser controlados por el usuario y un LED estático para indicación de funcionamiento. Los cuatro LED's verdes pueden ser controlados a través del SW mediante el manejo de los puertos GPIO.

#### **4.2.13 Entrada CTI JTAG.**

Una entrada para facilitar el desarrollo de SW y la depuración de la tarjeta con diversos emuladores JTAG. Con el fin de usar el conector, unas resistencias en serie deben ser removidas para aislar el USB a la función de JTAG.

#### **4.2.14 Interfases de expansión.**

Dos conectores de 46 pines de doble fila de 0,1 x 0,1 mm con conexión jack se suministran en el la tarjeta para el acceso a las señales de expansión. Debido al número de pines, un conector de baja fuerza de inserción ha sido elegido para facilitar la extracción de los cabos. Sin embargo, debido al gran número de pines, la extracción puede ser difícil y se debe tener cuidado cuando se la realiza.

#### **PERIFERICOS ADICIONALES QUE SE PUEDEN CONECTAR:**

#### **4.2.15 LCD.**

Un panel LCD completo de 24 bits puede ser conectado. Tiene luz de fondo y la funcionalidad de pantalla táctil, La potencia para la retroiluminación está limitada a 25 mA, por lo que se debe consultar el uso de paneles más grandes.

También se pueden conectar pantallas de LCD de 16 bits. La ventaja aquí es que esta utiliza menos patillas de los conectores de expansión dejando más señales para ser utilizadas por otras tarjetas de expansión.

#### **4.2.16 Puerto GPMC.**

Se proporciona acceso al bus GPMC. Dependiendo de la configuración necesaria, esto puede resultar en la pérdida de la interfaz LCD, pues no existiría la capacidad disponible, y limitará el uso a una pantalla LCD para 16Bits solamente.

#### **4.2.17 MMC1.**

A través de esta interfase se proporciona acceso directo a la tarjeta MMC.

#### **4.2.18 SPI.**

Hay dos puertos SPI (seriales) disponibles SPIO0 y SPIO1.

#### **4.2.19 I<sup>2</sup>C.**

Hay dos puertos I<sup>2</sup>C en el bus de expansión, I2C1 y I2C2. I2C2 se utiliza para las EEPROM sobre las tarjetas de expansión y siempre debe ser accesible. Se debe tener la precaución de no utilizar estas señales, pues otros componentes en un momento determinado pueden utilizar este bus.

#### **4.2.20 Puerto Serial.**

Hay 4 puertos serie con señales Tx, Rx, RTS, y CTS. Numerados como UART 1,2,4, el UART 5 solamente tiene Tx y Rx y el UART 3 no está disponible para el usuario.

#### **4.2.21 Conversores A/D.**

Siete conversores A/D de 100 Kmuestras por segundo, con un voltaje máximo de 1,8 Vcd.

#### **4.2.22 GPIO.**

Un máximo de 66 pines GPIO son accesibles desde los puertos de expansión. Todos estos son pines de 3,3 V y se pueden configurar como entradas o salidas digitales.

#### **4.2.23 CAN BUS.**

Hay dos interfaces CAN bus disponibles en la conexión de expansión BUS CAN versión 2 partes A y B. Las señales digitales RX y TX se identifican como se ve en la tabla al final.

#### **4.2.24 TIMERS.**

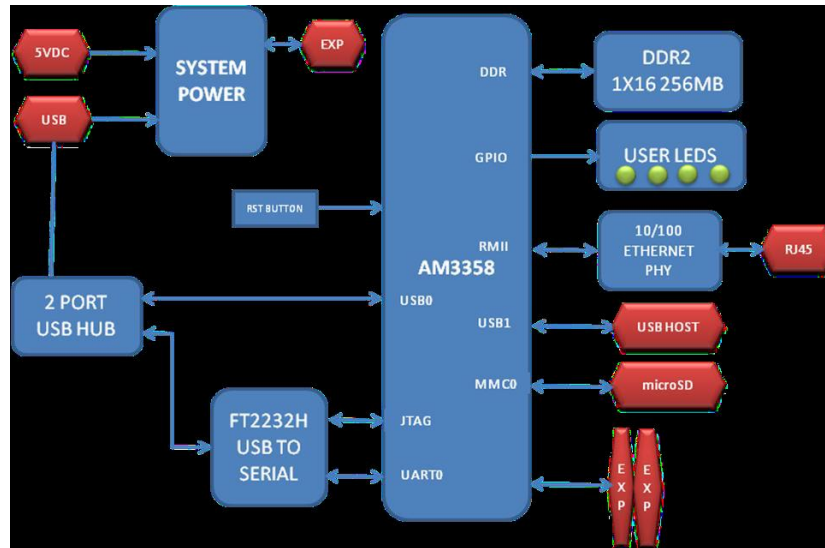
Cuatro temporizadores de alta velocidad.

#### **4.2.25 PWM.**

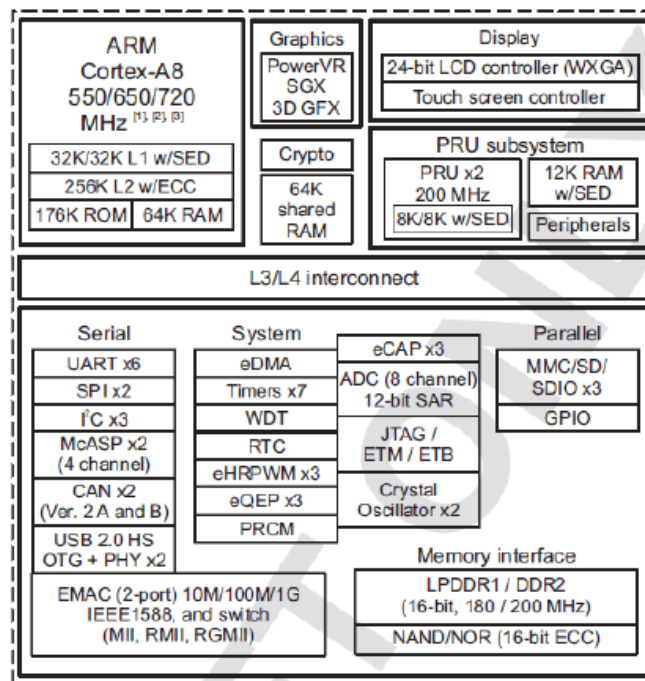
Ocho salidas PWM de alta resolución y modelo terminación simple.

A continuación en Figura 4.2 y Figura 4.3 los gráficos de la tarjeta y sus funcionalidades.





**Figura 4.2 Bloques constitutivos de la Beaglebone**



**Figura 4.3 Sistemas Componentes de la placa Beaglebone.**

Este tipo de sistemas empotrados pueden manejar diversos tipos o plataformas de sistema operativo, con la finalidad de poder realizar todas las operaciones a las que están sometidos, es lógico suponer que estos sistemas tienen capacidad de almacenamiento y operación limitada, no se puede pretender que a pesar de que son considerados ordenadores pequeños, que puedan tener un sistema operativo completo en su interior

incluso, con tareas de background, etc. Al contrario poseen sistemas operativos en tiempo real, de limitadas capacidades, orientadas justamente al manejo de sus aplicaciones.

Este es el caso de los sistemas operativos que se pueden ejecutar en Beaglebone, pues está desarrollada para ejecutar dos sistemas operativos muy distintos según las necesidades de los usuarios.

Por un lado puede instalarse un sistema operativo Android 4.0 orientado a realizar aplicaciones de interconexión de teléfonos inteligentes que usan el mismo sistema, que si se analiza fríamente el entorno, son la mayoría de Smart phones en la actualidad. En nuestro caso este sistema operativo no es una alternativa viable por el hecho de desconocerlo y las formas típicas de su manejo de las comunicaciones.

En segundo lugar puede también correr en la placa un sistema operativo Linux Ubuntu, que es el que se ha elegido y descargado por la funcionalidad que presenta para la elaboración de programas de usuario, que utilicen los recursos de la placa.

Este sistema operativo reside en una tarjeta micro SD y viene descargada una imagen de él desde la fábrica, pero por ser de distribución gratuita, se pueden descargar las actualizaciones más recientes de éste en la página de Beaglebone, que es [www.beaglebone.org](http://www.beaglebone.org), en donde se encuentra soporte del fabricante y muchos foros de discusión.

### **4.3 Método de desarrollo de software en la Beaglebone.**

El Kernel de Linux Ubuntu permite escribir programas con amplia funcionalidad para la placa Beaglebone, y el manejo de sus recursos hardware, es así que se pueden desarrollar aplicaciones en tres diferentes tipos de plataformas, la Primera y la más intuitiva es usando Python que puede manejar directamente los directorios y las opciones de las vías del kernel para hardware; es decir se comporta como una verdadera interfase de operación más que un lenguaje de programación. La segunda opción es utilizando JavaScript, al ser ya un lenguaje de programación, no una interfase de operación de hardware, se deben conocer las reglas de su escritura, sus sintaxis y sus códigos reservados. Y finalmente lenguaje C, que es la opción que se ha elegido, por dos condiciones específicas: permite manejar instrucciones alto nivel que manejen hardware y porque se lo conoce mejor que los demás y su programación es amigable y se pueden escribir programas en diferentes sistemas operativos y editores de texto y compilarlos directamente en Ubuntu.

Una de las principales ventajas de utilizar lenguaje C y C++ en el kernel del sistema operativo de la tarjeta Beaglebone, para la programación de las aplicaciones, es que no se necesita ningún compilador cruzado o cosa por el estilo, que opere independientemente

un programa editado por un editor de texto, fuera de la placa Beaglebone y después tenga que recibir para su ejecución algún código objeto de dicho programa, al contrario, como se ha mencionado anteriormente se puede trabajar de dos formas específicas con Beaglebone, la primera es conectarla una pantalla, teclado y un ratón y escribir directamente los programas en ella a través de una terminal de Ubuntu, compilarlos y ejecutarlos, como si tuviéramos una maquina desktop integrada. La segunda es escribir los programas fuente en un ordenador cualquiera y luego trasladarlos con sus propias características hacia la Beaglebone, con una simple instrucción de pegado, este programa se traslada a la placa y desde allí se lo puede compilar y ejecutar de forma sencilla y con las mismas instrucciones que las ejecutaría una maquina principal.

Esta opción es la que se ha elegido, pues da la posibilidad de escribir y compilar los programas en máquinas que ni siquiera estén utilizando el sistema operativo Linux, pero como la sintaxis de lenguaje C permite hacerlo, los podemos depurar en host distintos y remotos y trasladarlos ya revisados a la tarjeta Beaglebone y ejecutarlos, es decir trabajar solo con los programas objeto, administrando de mejor forma la capacidad de almacenamiento de la memoria SD de la placa.

Todo lo antes mencionado se ejecuta siguiendo el siguiente procedimiento:

En primer lugar existen dos formas de conexión y comunicación “principal” desde un ordenador a la placa Beaglebone:

- La primera consiste en hacer una conexión mediante un cable con conector USB en el lado del ordenador y conector mini USB en el lado de la Beagle, esta opción permitirá alimentar de voltaje a la Beagle a través del puerto USB del ordenador, es más sencilla, pero en contraparte, no están disponibles todas las características de la tarjeta, de hecho el procesador puede operar al 75% de su velocidad nada más.
- La segunda opción es, alimentar la Beagle a una fuente externa de 5 Vdc y conectar al ordenador mediante un cable directo de red Ethernet, sea directamente o sea a través de un router. Con esta opción aunque más complicada se obtienen 100% las prestaciones de la placa, en procesador y en todos sus sistemas de comunicaciones, además de tener disponibles salidas de hasta 1A de corriente proporcionados por la fuente externa.

Para trabajar en este proyecto se han utilizado las dos formas antes mencionadas, ya que en el conocimiento de la placa y las primeras investigaciones, lo más fácil era lo más adecuado, pero después se necesitaron todas las prestaciones del procesador, entonces se debió alimentar la placa con una fuente externa y configurar la red para usar la Beagle.

Hay que tomar en cuenta una cosa que es fundamental en este trabajo. El robot es Inalámbrico, es decir 100% autónomo, tanto en alimentación, como en comunicaciones y desarrollo de su software interno, es por eso que aquí, al final no importa cómo se hayan hecho los programas ni las conexiones, porque éstos corren en un robot autónomo.

Para la comunicación con la tarjeta desde un ordenador el procedimiento es el siguiente:

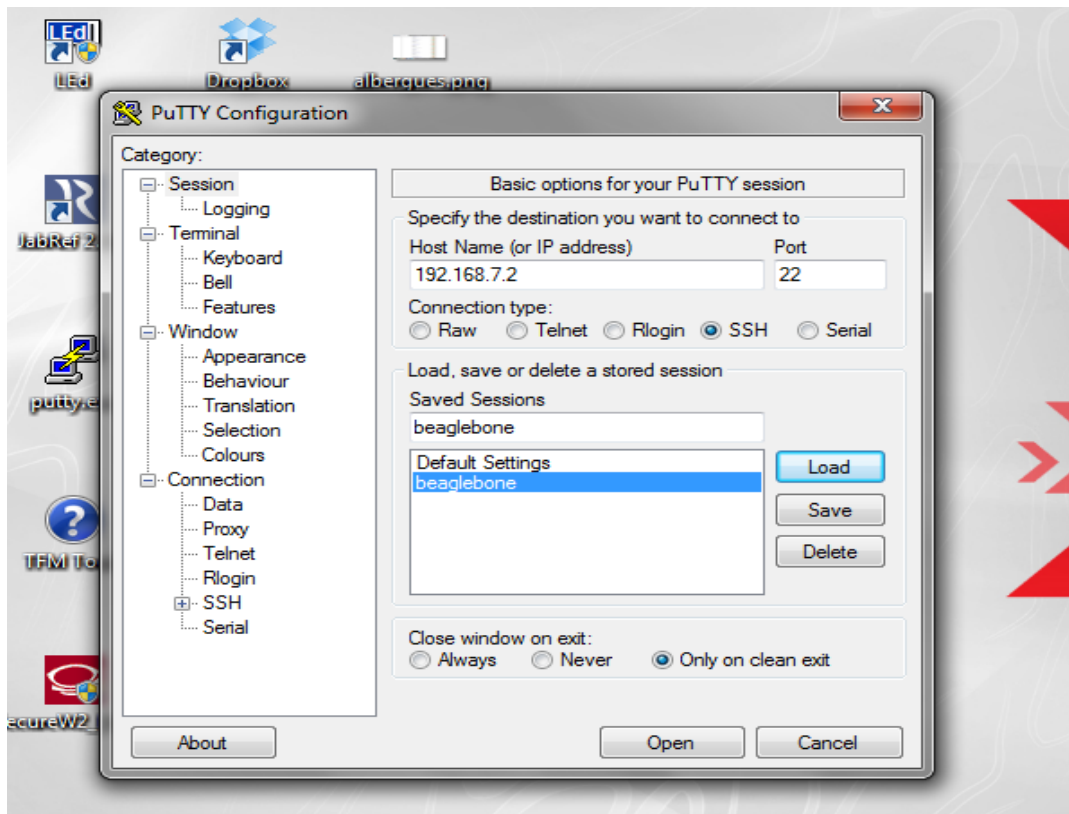
Independientemente el sistema operativo que su ordenador maneje, al conectar la Beaglebone a un ordenador, ésta aparecerá como un dispositivo de almacenamiento masivo, y de hecho allí se pueden observar la presencia de archivos importantes para el sistema, como los drivers que se utilizan para la misma conexión del puerto y en algunos casos los del sistema operativo que se ejecute en ella (Ubuntu o Android), pero no se pueden leer de ninguna manera los programas de usuario, que se han hecho en este caso en lenguaje C por ejemplo; en esta opción se pueden grabar programas de editores de textos, presentaciones, etc. Y trasladarlos, es decir se comporta como un penn memory común y corriente.

Para activarle a la placa se debe hacer click derecho en el mouse y expulsar el recurso.

Sin sacar el cable que conecta a las dos entidades se empieza a trabajar con la Beagle sin problemas. Ahora si está en dependencia del sistema operativo que se elija utilizar, pues si se decide trabajar en Windows, se deberá utilizar un terminal remoto, que administre una conexión Secure Shell (ssh), que permite a acceder a máquinas remotas, no tengo conocimiento de cuantas formas para hacer esto existan pero se ha utilizado una que es muy sencilla y que se llama PuTTY; PuTTY es un software dedicado que se lo puede obtener gratuitamente desde el siguiente enlace:

- <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Para el caso particular que se está tratando, la interfase que se ha utilizado es la siguiente:



**Figura 4.4 Conexión Putty para una Beaglebone en Windows 7.**

En la Figura 4.4 se puede observar la configuración de la conexión, es un protocolo de conexión de máquinas remotas a las cuales se acceden con la dirección del host que maneja y el puerto (serial) que se va a utilizar, en este caso la Beaglebone hace una conexión Ethernet, a través del puerto serial y PuTTY, utilizando su dirección IP que es la 192.168.7.2 y utiliza el puerto serial 22, que es la forma de comunicación USB que posee, es decir se enmascara una comunicación Ethernet a través del puerto USB y el terminal de PuTTY.

De esta forma se accede a la Beaglebone desde la terminal PuTTY, trabajando en sistema operativo Windows, como una terminal remota de Linux, ya que, se debe recordar que el sistema operativo que se ha descargado en la Beaglebone es Linux Ubuntu. Se deberán analizar particularidades como direcciones y claves de las tarjetas que se desee utilizar.

Otra forma, que es la que se ha elegido para el proyecto es utilizar, un ordenador que arranque directamente con sistema operativo Linux Ubuntu, y de allí ejecutar una terminal para acceder directamente a la Beaglebone de una forma parecida (es que se hace una conexión ssh también) que cuando se accedió a la conexión PuTTY.

Realizado este procedimiento se establece el servidor ssh directamente por la escritura del comando:

- `ssh root@192.168.7.2`

Es decir la dirección IP de la Beagle.

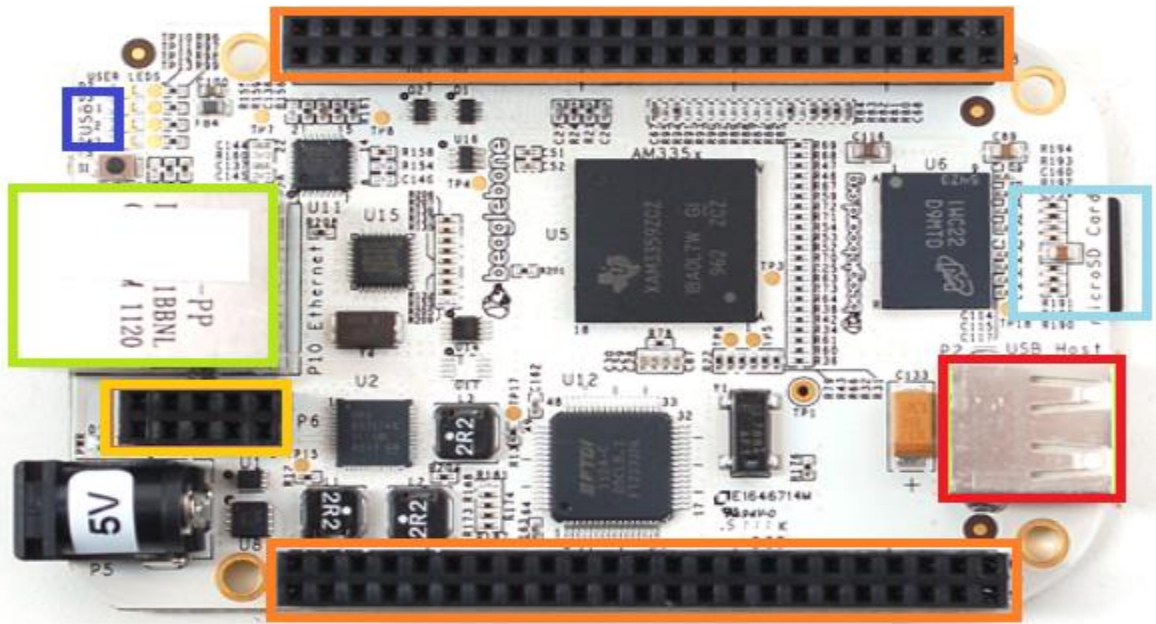
Se coloca el password que se ha establecido al bajar el kernel a la micro SD de la Beaglebone y a partir de ahí se maneja la consola dentro de la placa, como un ente de sistema operativo normal, en Linux Ubuntu; ahí se escribirán, compilarán y ejecutarán las aplicaciones de hardware.

Una ventaja de esta forma de proceder, es que se tiene a disposición los enormes recursos del ordenador para programar las aplicaciones, incluso transportarlas y migrarlas a otro S.O. y después de haber hecho todas las pruebas y depuraciones, se enviarán los programas finales dentro de la Beagle para que se compilen allí o solamente para que se ejecuten allí.

#### **4.4 Comunicaciones y puertos de expansión de la placa Beaglebone.**

Como se pudo apreciar en los párrafos anteriores, Beaglebone posee una serie de formas comunes de comunicación con el exterior, que van desde una simple comunicación serial, hasta redes y protocolos basados en patrones digitales que son usados a nivel industrial.

En la Figura 4.5 se presenta la placa Beaglebone vista desde su parte superior, aquí se enmascara la enorme capacidad de comunicaciones existentes, pues otras placas, como los arduinos, los igpv, los raspberry, etc. presentan muchos conectores específicos de cada puerto, por ejemplo el subD de 9 pines para el puerto serial, el USB A o B, RS485, etc. La placa Beaglebone solamente posee un Jack de RJ45 para la red Ethernet (resaltado en verde), dos jacks USB, uno tipo A (rojo) y uno tipo B (azul) y un lector de tarjetas micro SD (celeste). Los demás accesos a los medios de comunicaciones están en los dos header de 46 pines que posee en sus extremos (resaltados en color marrón), y que dan la posibilidad de personalizar la entrada del hardware que se necesite, ya que cada uno de estos pines no está dedicado ni personalizado para una sola entrada o salida, sino más bien se manejan 7 modos de operación de cada uno, controlados por la configuración software de un multiplexor central, además posee un conector de expansión de 10 pines (resaltado en amarillo).



**Figura 4.5 Entradas y Salidas de los diferentes Recursos en Beaglebone**

Los headers de entradas y salidas se han identificado como P8 y P9, siendo P8 el que está en la parte superior de la figura y P9 el otro. La descripción de la configuración disponible de cada uno de estos pines se puede observar en las siguientes tablas de descripción de cada uno de los puertos. En primer lugar los puertos en forma general, con los nombres que el fabricante ha decidido darle cada uno de los pines, y luego las posibilidades de operación de cada uno de ellos:

Tabla 4.2 Configuración general del Header P8 de Beglebone.

SIGNAL NAME	PROC	CONN		PROC	SIGNAL NAME
	GND	1	2	GND	
GPIO1_6	R9	3	4	T9	GPIO1_7
GPIO1_2	R8	5	6	T8	GPIO1_3
TIMER4	R7	7	8	T7	TIMER7
TIMER5	T6	9	10	U6	TIMER6
GPIO1_13	R12	11	12	T12	GPIO1_12
EHRPWM2B	T10	13	14	T11	GPIO0_26
GPIO1_15	U13	15	16	V13	GPIO1_14
GPIO0_27	U12	17	18	V12	GPIO2_1
EHRPWM2A	U10	19	20	V9	GPIO1_31
GPIO1_30	U9	21	22	V8	GPIO1_5
GPIO1_4	U8	23	24	V7	GPIO1_1
GPIO1_0	U7	25	26	V6	GPIO1_29
GPIO2_22	U5	27	28	V5	GPIO2_24
GPIO2_23	R5	29	30	R6	GPIO2_25
UART5_CTSN	V4	31	32	T5	UART5_RTSN
UART4_RTSN	V3	33	34	U4	UART3_RTSN
UART4_CTSN	V2	35	36	U3	UART3_CTSN
UART5_TXD	U1	37	38	U2	UART5_RXD
GPIO2_12	T3	39	40	T4	GPIO2_13
GPIO2_10	T1	41	42	T2	GPIO2_11
GPIO2_8	R3	43	44	R4	GPIO2_9
GPIO2_6	R1	45	46	R2	GPIO2_7



Tabla 4.3 Modos de operación 0-3 de los pines del header P8 de Beaglebone

PIN	PROC	NAME	MODE0	MODE1	MODE2	MODE3
1		GND				
2		GND				
3	R9	GPIO1_6	gpmc_ad6	mmc1_dat6		
4	T9	GPIO1_7	gpmc_ad7	mmc1_dat7		
5	R8	GPIO1_2	gpmc_ad2	mmc1_dat2		
6	T8	GPIO1_3	gpmc_ad3	mmc1_dat3		
7	R7	TIMER4	gpmc_advn_ale		timer4	
8	T7	TIMER7	gpmc_oen_ren		timer7	
9	T6	TIMER5	gpmc_be0n_cle		timer5	
10	U6	TIMER6	gpmc_wen		timer6	
11	R12	GPIO1_13	gpmc_ad13	lcd_data18	mmc1_dat5	mmc2_dat1
12	T12	GPIO1_12	GPMC_AD12	LCD_DATA19	MMC1_DAT4	MMC2_DAT0
13	T10	EHRPWM2B	gpmc_ad9	lcd_data22	mmc1_dat1	mmc2_dat5
14	T11	GPIO0_26	gpmc_ad10	lcd_data21	mmc1_dat2	mmc2_dat6
15	U13	GPIO1_15	gpmc_ad15	lcd_data16	mmc1_dat7	mmc2_dat3
16	V13	GPIO1_14	gpmc_ad14	lcd_data17	mmc1_dat6	mmc2_dat2
17	U12	GPIO0_27	gpmc_ad11	lcd_data20	mmc1_dat3	mmc2_dat7
18	V12	GPIO2_1	gpmc_clk_mux0	lcd_memory_clk	gpmc_wait1	mmc2_clk
19	U10	EHRPWM2A	gpmc_ad8	lcd_data23	mmc1_dat0	mmc2_dat4
20	V9	GPIO1_31	gpmc_csn2	gpmc_be1n	mmc1_cmd	
21	U9	GPIO1_30	gpmc_csn1	gpmc_clk	mmc1_clk	
22	V8	GPIO1_5	gpmc_ad5	mmc1_dat3		
23	U8	GPIO1_4	gpmc_ad4	mmc1_dat4		
24	V7	GPIO1_1	gpmc_ad1	mmc1_dat1		
25	U7	GPIO1_0	gpmc_ad0	mmc1_dat0		
26	V6	GPIO1_29	gpmc_csn0			
27	U5	GPIO2_22	lcd_vsync	gpmc_a8		
28	V5	GPIO2_24	lcd_pclk	gpmc_a10		
29	R5	GPIO2_23	lcd_hsync	gpmc_a9		
30	R6	GPIO2_25	lcd_ac_bias_en	gpmc_a11		
31	V4	UART5_CTSN	lcd_data14	gpmc_a18	eQEP1_index	mcasp0_axr1
32	T5	UART5_RTSN	lcd_data15	gpmc_a19	eQEP1_strobe	mcasp0_ahclkx
33	V3	UART4_RTSN	lcd_data13	gpmc_a17	eQEP1B_in	mcasp0_fsr
34	U4	UART3_RTSN	lcd_data11	gpmc_a15	ehrpwm1B	mcasp0_ahclkx
35	V2	UART4_CTSN	lcd_data12	gpmc_a16	eQEP1A_in	mcasp0_aclkr

PIN	PROC	NAME	MODE0	MODE1	MODE2	MODE3
36	U3	UART3_CTSN	lcd_data10	gpmc_a14	ehrpwm1A	mcasp0_axr0
37	U1	UART5_TXD	lcd_data8	gpmc_a12	ehrpwm1_tripzone_in	mcasp0_aclkx
38	U2	UART5_RXD	lcd_data9	gpmc_a13	ehrpwm0_synco	mcasp0_fsx
39	T3	GPIO2_12	lcd_data6	gpmc_a6		eQEP2_index
40	T4	GPIO2_13	lcd_data7	gpmc_a7		eQEP2_strobe
41	T1	GPIO2_10	lcd_data4	gpmc_a4		eQEP2A_in
42	T2	GPIO2_11	lcd_data5	gpmc_a5		eQEP2B_in
43	R3	GPIO2_8	lcd_data2	gpmc_a2		ehrpwm2_tripzone_in
44	R4	GPIO2_9	lcd_data3	gpmc_a3		ehrpwm0_synco
45	R1	GPIO2_6	lcd_data0	gpmc_a0		ehrpwm2A
46	R2	GPIO2_7	lcd_data1	gpmc_a1		ehrpwm2B

**Tabla 4.4 Modos de operación 4-7 de los pines del header P8 de Beaglebone**

PIN	PROC	NAME	MODE4	MODE5	MODE6	MODE7
1		GND				
2		GND				
3	R9	GPIO1_6				gpio1[6]
4	T9	GPIO1_7				gpio1[7]
5	R8	GPIO1_2				gpio1[2]
6	T8	GPIO1_3				gpio1[3]
7	R7	TIMER4				gpio2[2]
8	T7	TIMER7				gpio2[3]
9	T6	TIMER5				gpio2[5]
10	U6	TIMER6				gpio2[4]
11	R12	GPIO1_13	eQEP2B_in			gpio1[13]
12	T12	GPIO1_12	EQEP2A_IN			gpio1[12]
13	T10	EHRPWM2B	ehrpwm2B			gpio0[23]
14	T11	GPIO0_26	ehrpwm2_tripzone_in			gpio0[26]
15	U13	GPIO1_15	eQEP2_strobe			gpio1[15]
16	V13	GPIO1_14	eQEP2_index			gpio1[14]
17	U12	GPIO0_27	ehrpwm0_synco			gpio0[27]
18	V12	GPIO2_1			mcasp0_fsr	gpio2[1]
19	U10	EHRPWM2A	ehrpwm2A			gpio0[22]
20	V9	GPIO1_31				gpio1[31]

21	U9	GPIO1_30				gpio1[30]
22	V8	GPIO1_5				gpio1[5]
23	U8	GPIO1_4				gpio1[4]
24	V7	GPIO1_1				gpio1[1]
25	U7	GPIO1_0				gpio1[0]
26	V6	GPIO1_29				gpio1[29]
27	U5	GPIO2_22				gpio2[22]
28	V5	GPIO2_24				gpio2[24]
29	R5	GPIO2_23				gpio2[23]
30	R6	GPIO2_25				gpio2[25]
31	V4	UART5_CTSN	uart5_rxd		uart5_ctsn	gpio0[10]
32	T5	UART5_RTSN	mcasp0_axr3		uart5_rtsn	gpio0[11]
33	V3	UART4_RTSN	mcasp0_axr3		uart4_rtsn	gpio0[9]
34	U4	UART3_RTSN	mcasp0_axr2		uart3_rtsn	gpio2[17]
35	V2	UART4_CTSN	mcasp0_axr2		uart4_ctsn	gpio0[8]
36	U3	UART3_CTSN			uart3_ctsn	gpio2[16]
37	U1	UART5_TXD	uart5_txd		uart2_ctsn	gpio2[14]

PIN	PROC	NAME	MODE4	MODE5	MODE6	MODE7
38	U2	UART5_RXD	uart5_rxd		uart2_rtsn	gpio2[15]
39	T3	GPIO2_12				gpio2[12]
40	T4	GPIO2_13	pr1_edio_data_out7			gpio2[13]
41	T1	GPIO2_10				gpio2[10]
42	T2	GPIO2_11				gpio2[11]
43	R3	GPIO2_8				gpio2[8]
44	R4	GPIO2_9				gpio2[9]
45	R1	GPIO2_6				gpio2[6]
46	R2	GPIO2_7				gpio2[7]

Tabla 4.5 Configuración general de los pines del header P9 de Beaglebone

SIGNAL NAME	PIN	CONN	PIN	SIGNAL NAME
	GND	1	2	GND
	VDD_3V3EXP	3	4	VDD_3V3EXP
	VDD_5V	5	6	VDD_5V
	SYS_5V	7	8	SYS_5V
PWR_BUT*		9	10	A10
UART4_RXD	T17	11	12	U18
UART4_TXD	U17	13	14	U14
GPIO1_16	R13	15	16	T14
I2C1_SCL	A16	17	18	B16
I2C2_SCL	D17	19	20	D18
UART2_TXD	B17	21	22	A17
GPIO1_17	V14	23	24	D15
GPIO3_21	A14	25	26	D16
GPIO3_19	C13	27	28	C12
SPI1_D0	B13	29	30	D12
SPI1_SCLK	A13	31	32	VDD_ADC(1.8V)
AIN4	C8	33	34	GNDA_ADC
AIN6	A5	35	36	A5
AIN2	B7	37	38	A7
AIN0	B6	39	40	C7
CLKOUT2	D14	41	42	C18
	GND	43	44	GND
	GND	45	46	GND

Tabla 4.6 Modos de operación 0-3 de los pines del header P9 de Beaglebone

PIN	PROC	SIGNAL NAME	MODE0	MODE1	MODE2	MODE3
1		GND				
2		GND				
3		DC_3.3V				
4		DC_3.3V				
5		VDD_5V				
6		VDD_5V				
7		SYS_5V				
8		SYS_5V				
9		PWR_BUT				
10	A10	SYS_RESETn	RESET_OUT			
11	T17	UART4_RXD	gpmc_wait0	mii2_crs	gpmc_csn4	rmii2_crs_dv
12	U18	GPIO1_28	gpmc_be1n	mii2_col	gpmc_csn6	mmc2_dat3
13	U17	UART4_TXD	gpmc_wpn	mii2_rxerr	gpmc_csn5	rmii2_rxerr
14	U14	EHRPWM1A	gpmc_a2	mii2_txd3	rgmii2_txd3	mmc2_dat1
15	R13	GPIO1_16	gpmc_a0	gmii2_txen	rmii2_tctl	mii2_txen
16	T14	EHRPWM1B	gpmc_a3	mii2_txd2	rgmii2_txd2	mmc2_dat2
17	A16	I2C1_SCL	spi0_cs0	mmc2_sdwp	I2C1_SCL	ehrpwm0_synci
18	B16	I2C1_SDA	spi0_d1	mmc1_sdwp	I2C1_SDA	ehrpwm0_tripzone
19	D17	I2C2_SCL	uart1_rtsn	timer5	dcan0_rx	I2C2_SCL
20	D18	I2C2_SDA	uart1_ctsn	timer6	dcan0_tx	I2C2_SDA
21	B17	UART2_TXD	spi0_d0	uart2_txd	I2C2_SCL	ehrpwm0B
22	A17	UART2_RXD	spi0_sclk	uart2_rxd	I2C2_SDA	ehrpwm0A
23	V14	GPIO1_17	gpmc_a1	gmii2_rxdv	rgmii2_rxdv	mmc2_dat0
24	D15	UART1_TXD	uart1_txd	mmc2_sdwp	dcan1_rx	I2C1_SCL
25	A14	GPIO3_21	mcasp0_ahclkx	eQEP0_strobe	mcasp0_axr3	mcasp1_axr1
26	D16	UART1_RXD	uart1_rxd	mmc1_sdwp	dcan1_tx	I2C1_SDA
27	C13	GPIO3_19	mcasp0_fsr	eQEP0B_in	mcasp0_axr3	mcasp1_fsx
28	C12	SPI1_CS0	mcasp0_ahclkx	ehrpwm0_synci	mcasp0_axr2	spi1_cs0
29	B13	SPI1_D0	mcasp0_fsx	ehrpwm0B		spi1_d0
30	D12	SPI1_D1	mcasp0_axr0	ehrpwm0_tripzone		spi1_d1
31	A13	SPI1_SCLK	mcasp0_aclkx	ehrpwm0A		spi1_sclk
32		VADC				
33	C8	AIN4				
34		AGND				
35	A8	AIN6				

PIN	PROC	SIGNAL NAME	MODE0	MODE1	MODE2	MODE3
36	B8	AIN5				
37	B7	AIN2				
38	A7	AIN3				
39	B6	AIN0				
40	C7	AIN1				
41	D14	CLKOUT2	xdma_event_intr1		tdlkin	clkout2
42	C18	GPIO0_7	ecap0_in_PWM0_out	uart3_txd	spi1_cs1	pr1_ecap0_ecap_cap_in_apwm_o
43		GND				
44		GND				
45		GND				
46		GND				

**Tabla 4.7 Modos de operación 4-7 de los pines del header 9 de Beaglebone**

PIN	PROC	SIGNAL NAME	MODE4	MODE5	MODE6	MODE7
1		GND				
2		GND				
3		DC_3.3V				
4		DC_3.3V				
5		VDD_5V				
6		VDD_5V				
7		SYS_5V				
8		SYS_5V				
9		PWR_BUT				
10	A10	SYS_RESETn				
11	T17	UART4_RXD	mmc1_sdcd		uart4_rxd_mux2	gpio0[30]
12	U18	GPIO1_28	gpmc_dir		mcasp0_aclkr_mux3	gpio1[28]
13	U17	UART4_TXD	mmc2_sdcd		uart4_txd_mux2	gpio0[31]
14	U14	EHRPWM1A	gpmc_a18		ehrpwm1A_mux1	gpio1[18]

15	R13	GPIO1_16	gpmc_a16		ehrpwm1_tripzone_input	gpio1[16]
16	T14	EHRPWM1B	gpmc_a19		ehrpwm1B_mux1	gpio1[19]
17	A16	I2C1_SCL				gpio0[5]
18	B16	I2C1_SDA				gpio0[4]
19	D17	I2C2_SCL	spi1_cs1			gpio0[13]
20	D18	I2C2_SDA	spi1_cs0			gpio0[12]
21	B17	UART2_TXD			EMU3_mux1	gpio0[3]
22	A17	UART2_RXD			EMU2_mux1	gpio0[2]
23	V14	GPIO1_17	gpmc_a17		ehrpwm0_synco	gpio1[17]
24	D15	UART1_TXD				gpio0[15]
25	A14	GPIO3_21	EMU4_mux2			gpio3[21]
26	D16	UART1_RXD				gpio0[14]
27	C13	GPIO3_19	EMU2_mux2			gpio3[19]
28	C12	SPI1_CS0	eCAP2_in_PWM2_out			gpio3[17]
29	B13	SPI1_D0	mmc1_sdcd_mux1			gpio3[15]
30	D12	SPI1_D1	mmc2_sdcd_mux1			gpio3[16]
31	A13	SPI1_SCLK	mmc0_sdcd_mux1			gpio3[14]
32		VDD_ADC (1.8V_				
33	C8	AIN4				
34		GNDA_ADC				
35	A5	AIN6				

PIN	PROC	SIGNAL NAME	MODE4	MODE5	MODE6	MODE7
36	A5	AIN5				
37	B7	AIN2				
38	A7	AIN3				
39	B6	AIN0				
40	C7	AIN1				
41	D14	CLKOUT2	timer7_mux1		EMU3_mux0	gpio0[20]
42	C18	GPIO0_7	spi1_sclk	mmc0_sdwp	xdma_event_intr2	gpio0_7
43		GND				
44		GND				
45		GND				
46		GND				



Como se puede observar al combinar todos los pines existentes en los header de la placa Beaglebone, las posibilidades son muy grandes, de tal forma que los mismos diseñadores de la placa se dan cuenta que sería imposible encontrar una aplicación en la que todas las opciones fueran utilizadas, es por esto que se utilizan los modos distintos de operación del header. Además de que se pueden utilizar directamente los pines correspondientes, el fabricante da la posibilidad de utilizar recursos estándar de los sistemas como conectores e indicadores, es decir sus modos de entrada y salida estándar, esto se lo realiza mediante la adquisición de interfases dedicadas para ello, conocidas como “capex”, que se conectan en forma directa y muy fácil a los header, con la utilización de conectores de 46 pines machos. Más información se la puede encontrar en: <http://beagleboard.org/cape>.

La forma en que se programan cada uno de los modos de operación, es entrando en la plataforma, con el método antes mencionado, y modificar manualmente el modo de operación, mandando el número equivalente al que se desee usar. De la siguiente manera:

- En el directorio principal de Linux buscar `$ sys/kernel/debug/omap_mux`
- Ahí se pueden ver todos los pines configurables y con la instrucción. `$ cat xxxx`, donde xxxx es el nombre del pin necesitado, se ve el modo en el que está; por defecto la mayoría se encuentran en modo 0, pero hay que verificarlo para no cometer errores.
- Por ejemplo para llevarle a modo 7 al GPIO1\_6 que es el pin 3 del header P8, que tiene ese mismo nombre, simplemente se escribe: `$ echo 07 > gpio1_6`, o bien `$ echo 27 > gpio1_6`, que dará el mismo resultado.
- A partir de ahí dependerá de qué tipo sea la entrada o salida que se ha elegido para hacer las pruebas y configuraciones particularidades, por ejemplo en un gpio se puede configurar su dirección (entrada o salida) de la siguiente manera: `$ echo out > direction`.

## 4.5 Placa Adaptadora.

Como ya se mencionó anteriormente, aquí ve va a integrar muchos sistemas que manejan protocolos estándar de comunicaciones, pero no los mismos niveles de tensión, para que su funcionamiento sea adecuado, se deben adaptar todos esos voltajes y señales de tierra correspondientes. Para ello se hace necesaria la construcción de una tarjeta adicional que adapte todas las señales a utilizarse, como este es un prototipo personalizado de robot, se diseñó esta placa a partir de los verdaderos requisitos del sistema.



En primer lugar es necesaria la correcta definición de todos los elementos que se van a necesitar, desde componentes electrónicos, hasta conectores y sujetadores que los van a contener.

Para esto se ha utilizado el software Altium Designer. Este es un paquete o conjunto de programas que se utilizan para el diseño electrónico en cada una de sus fases, es decir se puede emplear para obtener esquemas, simulación y diseño de circuitos impresos. Para este trabajo se lo ha utilizado solamente con el esquema para poder crear el PCB o la placa, no se utilizaron sus capacidades de simulación.

La primera característica que fue necesaria tomar en cuenta, fue los componentes externos, que dan y reciben señales de la placa Beaglebone, y sus requerimientos, que son los siguientes:

- La placa necesita manejar motores de cc (no paso a paso) con eficiencia (velocidad y torque), haciéndole variar la velocidad a estos motores mediante el valor medio entregado por una señal digital que controla un sistema PWM.
- La placa necesita saber cuánto y cómo se ha movido, para ello se utilizan encoders incrementales en cada una de las ruedas, estas señales deben introducirse a puertos digitales de entrada de la Beagle, verificando en ella, la fase y el desplazamiento de las señales que entregan los encoders.
- La placa necesita conectarse con una nariz electrónica mediante el protocolo I<sup>2</sup>C, que generalmente opera con señales de 1.8Vdc, y estas introducirlas a la Beaglebone en el sistema I<sup>2</sup>C que maneja 3,3 Vdc.
- La placa necesita comunicarse con los demás robots en conexión de red, para lo que se seleccionó el sistema XBee, que utiliza niveles de polarización de 3,3Vdc y niveles de entrada y salida similares. Que se adaptan de forma directa con la Beagle, que los maneja de la misma forma.
- La placa necesita saber si en el trayecto del robot hay obstáculos, para lo que se le han conectado 4 sensores infrarrojos de posición que presentan una respuesta analógica de hasta 5 Vdc, pero la entrada analógica de la placa Beaglebone, solo acepta niveles de voltaje de hasta 1,8 vdc.
- No hay que olvidarse que el robot es autónomo, y que tiene que funcionar con una batería que sustente 100% su operación. Pero hay que considerar que las pruebas no se pueden hacer con la batería conectada permanentemente, sino más bien con un cargador electrónico.

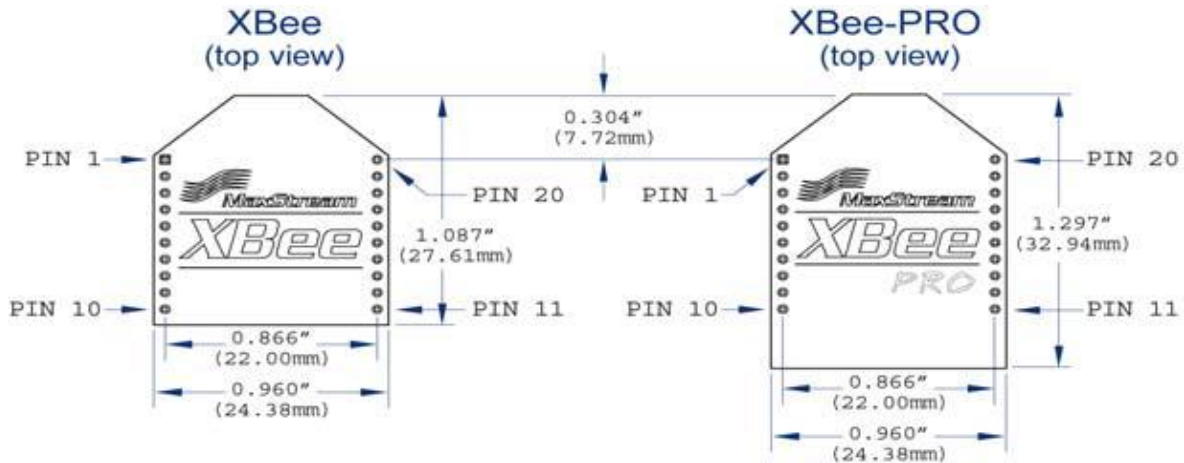
#### 4.6 Diseño de la Placa.

El diseño de la placa se basa en dos componentes principales, que determinarán la forma y posición óptima de los demás elementos. En primer lugar la placa Beaglebone que va ir conectada en la parte inferior de la placa auxiliar, sin ningún otro mecanismo que los dos conectores header que tiene la Beagle, que la soportan sin problemas por su reducido peso.

[illegible]

74

Pero, a pesar que la placa Beaglebone es el propósito principal de la placa madre, también existen muchos otros elementos que poseen conectores especiales, como el módulo XBee, que también debe tener dimensiones específicas para poder utilizarlo fuera de una de sus placas madre. Esto se observa en la Figura 4.7



**Figura 4.7 Dimensiones de los módulos XBee.**

Esto significa, que se tienen que tomar en cuenta primero estas dimensiones y a partir de ahí colocar los demás objetos en la placa.

Los demás “puertos” de entrada y salida a la placa auxiliar, no tienen modelo específico, por lo que se decidió conectarlos por conectores Molex de diversas configuraciones (2,3,4 y 6 pines por conector). Esto es una ventaja porque las librerías de altium ya los soportan al contrario de XBee y otros.

Entonces el primer elemento a utilizarse es un header de 2x24 pines, ya que no existían conectores de 2x23, pero se coloca el footprint de este mismo dejando los dos últimos agujeros sin conectar.

Luego de ello se verifico la existencia del header para la placa XBee, pero no existía por lo que se utilizaron dos header de 10 pines cada uno, y se construyó el footprint de 20 pines, que se asignó a uno de los header de 10 pines que se eligió.

Luego dos conectores molex de 2 vías para la conexión de la batería que sustente al robot y para entregar tensión de ser necesaria a una placa adicional que se pueda conectar arriba o abajo, o sacar hacia otro sitio polarización regulada de 5 Vdc. 4 conectores molex de tres pines para la conexión de los 4 sensores de posición, 2 conectores de 6 pines para los motores (uno por cada motor), que en conjunto tienen las señales de cada terminal del motor, la polarización para la placa de los encoders y la salida de cada uno de los encoders. Finalmente un conector molex de 4 pines para la nariz electrónica.

A partir de allí se utilizaron los elementos necesarios para las alimentaciones y acoplamientos de cada uno de los elementos de la placa adaptadora.

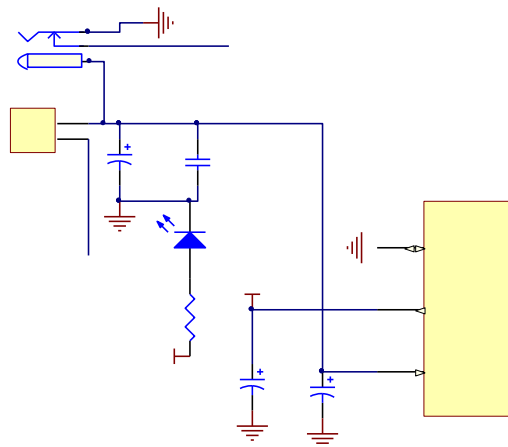
## 4.7 Fuente de alimentación.

Al ser muchos los componentes que se van a utilizar y de diversos niveles de tensión, se debe elegir una tensión de alimentación para todos ellos, además de la Beaglebone fundamentalmente, es por esta razón que se escogió un voltaje regulado de 5 Vdc, para alimentar la Beagle, al mismo tiempo de que se administren todos los demás voltajes necesarios que son:

- 5Vdc para alimentación de la Beagle y los sensores de posición, los encoders y los motores.
- 3,3Vdc para los módulos XBee y los puertos de I/O de Beaglebone.
- 1.8Vdc para la nariz electrónica y las señales ADC.

Vistos estos requisitos se decidió utilizar el regulador de voltaje TLV1117 de TI, por las características eléctricas y mecánicas que presenta, su pequeño tamaño, su característica de montaje superficial, que tiene muchos niveles de conversión de tensión, es decir no es fijo, y su corriente de salida es bastante aceptable para el número de cargas que se van a conectar; además de que al robot se lo va a alimentar, vista su autonomía con una batería portátil ultra eficiente de 9 voltios, y éste regulador acepta voltajes de hasta 16Vdc.

En la Figura 4.8 se observa el esquemático de conexión de la fuente de alimentación utilizando el regulador de tensión TLV1117, en la cual se observa su sencillez de conexión y los parámetros necesarios para su funcionamiento.



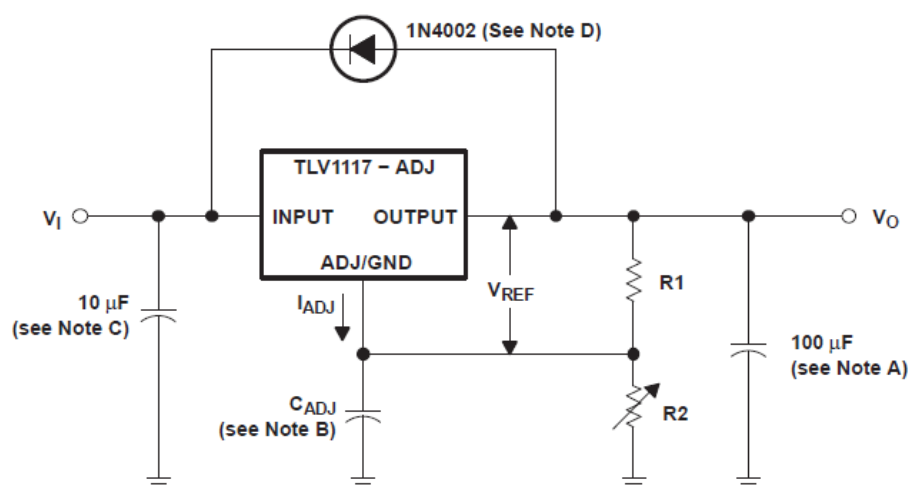
**Figura 4.8 Esquemático de la fuente de alimentación de la placa adaptadora**

En ella se pueden observar ciertos detalles, como que la entrada puede ser obtenida de dos formas, mediante el header molex de dos pines, que sería la señal proveniente de la batería portátil, y a través de un conector Jack, que permite que se le conecte también una fuente electrónica cableada, la razón de esto es que no se puede utilizar permanentemente la batería para las pruebas porque se descargaría, entonces en estos casos se utiliza la fuente electrónica. Y segundo que la entrada del regulador está planificada de 5 a 9Vdc, es por el hecho de que las baterías se descargan y se puede utilizar una que tenga un poco más de 5Vdc de carga y los cargadores electrónicos de pared vienen en diferentes configuraciones: en 5, 6 7 9 12Vdc, pudiéndose entonces utilizar uno de hasta 9Vdc.

Se ha colocado un led monitor de alimentación a la salida del regulador, que indicará su funcionamiento, y los capacitores han sido los que el fabricante del regulador recomienda en su datasheet.

El regulador de voltaje **TLV1117**, es un circuito que puede proporcionar tensiones fijas de salida de: 1.5-V, 1.8-V, 2.5-V, 3.3-V y 5V con una corriente de 800mA, además de voltajes regulables. A él se le pueden alimentar tensiones fijas de cc de hasta 16 V, y configurarlo para la tensión de salida fija requerida o tensiones variables.

En este caso se le ha programado para que emita una señal de salida fija de 5Vdc para una entrada de 5 a 9Vdc, con el siguiente circuito obtenido de sus especificaciones:



**Figura 4.9 Circuito regulador de voltaje.**

Al no desear que el voltaje de salida sea variable se ha conectado la señal ADJ/GND directamente a tierra.

## 4.8 Motores y encoders.

El robot está provisto de dos ruedas motoras y dos ruedas “locas” que soportan la parte frontal y trasera del mismo. Las ruedas que tienen tracción, están conectadas a sendos motores de relaciones muy altas y considerable potencia y torque, que se pueden acoplar fácilmente a diversos tipos de ruedas, así como a una placa que contienen dos encoders incrementales, que sirven para determinar el paso del motor según los diétes que poseen las ruedas en su interior.

El conjunto utilizado de motores, ruedas y encoders, corresponden a la serie Pololu 2218 del motor, ruedas de caucho de 42X19 mm y encoder, se muestran en la Figura 4.10.



**Figura 4.10 Motores, Rueda y Encoders Pololu.**

Como se puede apreciar, el motor tiene la posibilidad de avanzar a favor o en contra de las manecillas del reloj, según la conexión que se le proporciona de la fuente de corriente constante, además en la placa se visualizan también dos encoders incrementales colocados dentro del aro de la rueda, y que están diseñados y montados para leer los dientes de la parte interior de la rueda y así determinar su paso, al ser dos encoders se necesita una salida para cada uno de ellos y la alimentación para la placa, que es de 5Vdc.

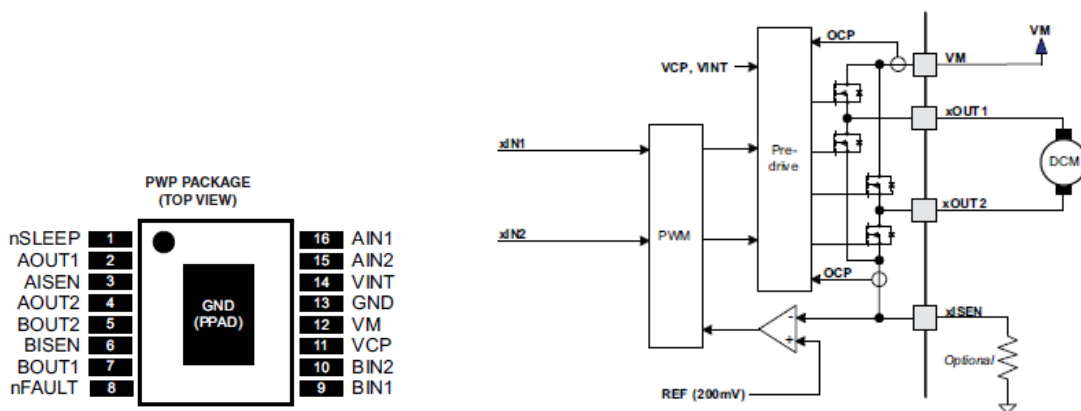
Esto nos hace ver que se debe enviar desde la placa adaptadora la polarización para la placa de los encoders y recibir sus señales que serán de la misma magnitud de voltaje es decir 5Vdc, al mismo tiempo que proporcionarle el voltaje necesario para que avancen las ruedas del robot accionadas por los motores.

Al no poseer una señal de salida analógica en voltaje en la Beaglebone, es difícil alimentar solamente con una sola polaridad al motor y después cambiarla dándole la vuelta a sus terminales o algo parecido, se deben encontrar alternativas de utilización, y es así que se ha empleado las señales de salida provenientes de un control PWM que posee la placa

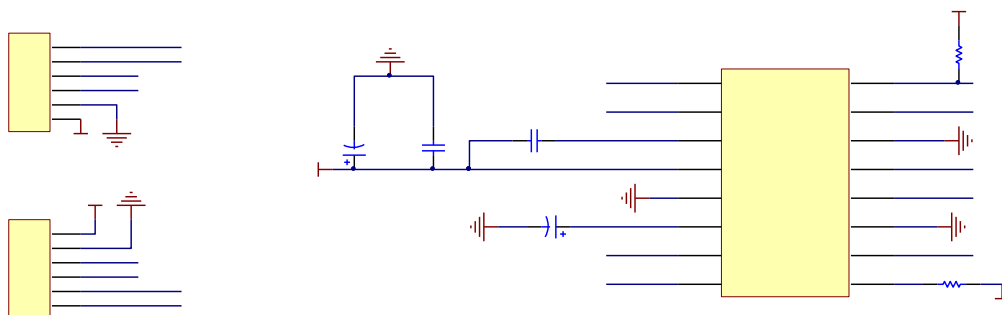
Beaglebone, que además de proporcionar una variación analógica de una salida digital, a una frecuencia determinada, nos permite invertir de giro al motor con cambios en software nada más.

Las señales de los headers de Beaglebone tanto para los GPIO como para muchas otras, manejan niveles de tensión de 3.3Vdc, que no es suficiente para conseguir una eficiente acción de los motores, es por esta razón que se ha empleado un circuito adicional para conseguir esta adaptación, en un circuito integrado que controla cada motor a través de un puente H, dándole mayor versatilidad y fuerza, ya que tiene la capacidad de entregar hasta 1,A de corriente, con el consumo de potencias relativamente bajas. El circuito se basa en el diseño de José María Martín Talavera, empleado en su trabajo de fin de carrera en el 2012.

Se utiliza el C.I. DRV8833 de T.I., que contiene en su interior dos puentes H, que permiten girar al motor en ambos sentidos, se puede polarizar hasta 10Vdc, pero como la placa adaptadora lleva una tensión adre de 5Vdc, se le ha polarizado con ese valor. En la Figura 4.11. se indica la distribución de pines y el circuito utilizado en la placa adaptadora.



**Figura 4.11 Diagrama de distribución de pines y circuito recomendado para uso de PWM. Extraído del datasheet del fabricante Texas Instruments.**



**Figura 4.12 Diagrama de conexiones del driver de los motores en Altium Designer.**

Como se puede observar en la Figura 4.12, tiene dos entradas para cada motor signadas como AIN1 y AIN2, cuyas correspondientes salidas son AOUT1 y AOUT2, lo mismo para el puente B. La señal NSLEEP sirve para habilitar el driver, NFAULT para indicar sobrecarga y calentamiento del circuito, las señales de control pueden ser enviadas en niveles de voltaje lógico entre cero y uno, o como una señal PWM de acuerdo a la lógica presentada en las siguientes tablas, obtenidas del datasheet del fabricante.

**Tabla 4.8 Lógica de funcionamiento del puente H.**

xIN1	xIN2	xOUT1	xOUT2	FUNCTION
0	0	Z	Z	Coast/fast decay
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	L	L	Brake/slow decay

**Tabla 4.9 Control PWM de la velocidad del motor.**

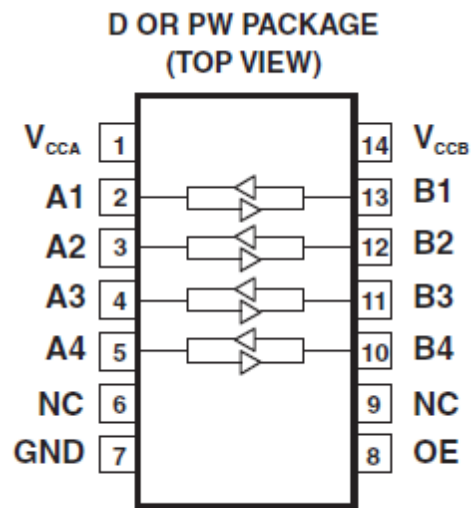
xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay

En este proyecto se utilizará la tabla PWM de rápido decaimiento, por las condiciones de velocidad que necesita el motor. Finalmente se puede ver en la distribución de pines que se necesita un plano de masa conectado detrás del circuito y que puede servir como mecanismo de disipación de calor.

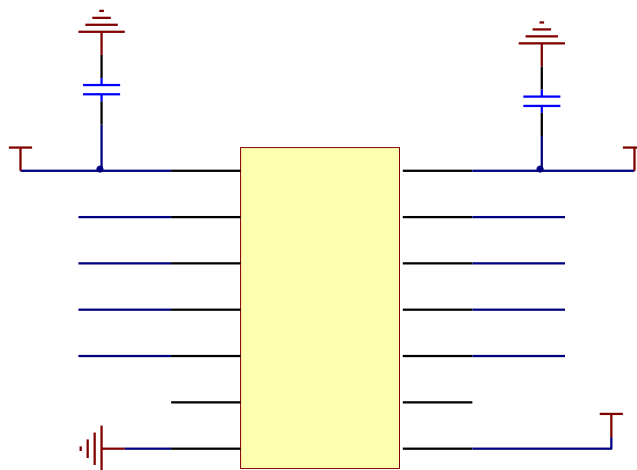


En el diagrama de altium designer, se mencionan las conexiones con la placa Beaglebone y se explicarán más adelante.

Los encoders en cambio poseen dos entradas de alimentación de 5Vdc, y presentan a la salida dos líneas que corresponden al encoder A y al B. Recordando que Beaglebone solo acepta entradas de 3.3Vdc, se debe utilizar un circuito adaptador de niveles de voltaje, este es el buffer TXB0104 de Texas Instruments. En la Figura 4.13 se presenta su distribución de pines y sus conexiones en altium designer:



**Figura 4.13 Distribución circuital de TXB0104, obtenido del datasheet del fabricante.**



**Figura 4.14 Conexión de los pines del circuito buffer en Altium Designer.**

Como se observa en la Figura 4.14 es un buffer bidireccional, sin control de dirección, que convierte la tensión de entrada en su equivalente de salida en un voltaje menor o mayor. Su funcionamiento se basa en la polarización de ambos lados, pudiendo ser de 1.2 a 3.6Vdc en el puerto A y de 1.65 a 5.5Vdc en el puerto B, con la consideración que siempre  $V_{CCB}$  debe ser mayor que  $V_{CCA}$ .

En este circuito cada línea es independiente y no hace falta indicarle la dirección de entrada de la corriente, lo detecta automáticamente y lo convierte en su lado de salida. El requerimiento es que Todas las líneas del lado A coincidan en voltaje con su referencia  $V_{CCA}$ , lo mismo que las líneas del lado B con su referencia  $V_{CCB}$ .

De esta forma se acoplan perfectamente las señales que provienen de los encoders y que vienen de una placa polarizada con 5Vdc a las entradas de la placa Beaglebone que permiten máximo una entrada de 3.3Vdc (los buffer solamente convierten los valores digitales de dichas señales, es decir 3.3 a 5 o 5 a 3.3, para señales análogas se podría construir un seguidor de voltaje con tensiones acopladas. En este diseño no hay problema porque las señales de los encoders son pulsos digitales.

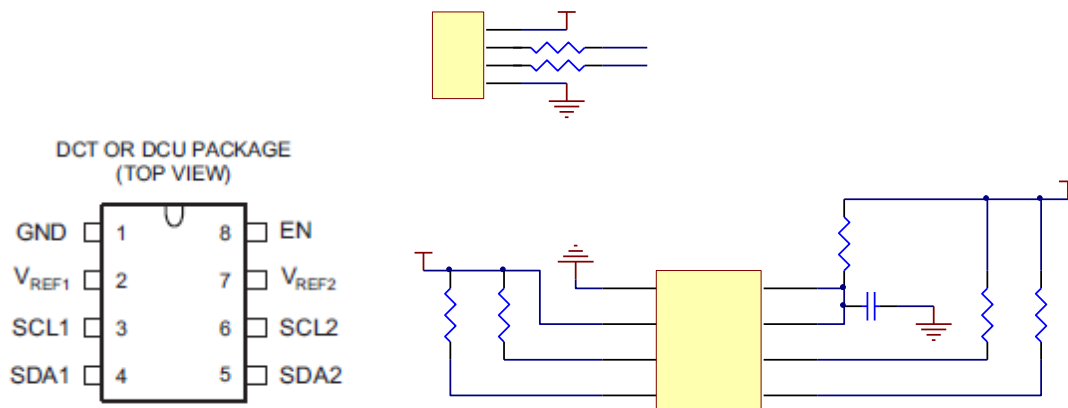
Estos buffer se utilizarán tanto para el ingreso de las señales de los encoders al header de Beaglebone, como para el ingreso de las señales de los sensores de obstáculos (SHARP, que no se acondicionará porque no hay ese requisito, tampoco se implementará el software por no ser parte de la colaboración, pero se dejará el hardware perfectamente instalado para que se conecten los sensores a los conectores molex y se escriba la función de software e introduzca en el programa principal).

## **4.9 Conexión de la Nariz electrónica.**

El grupo de investigación GNB, tiene ya desarrollado un módulo de nariz electrónica que entrega una señal digital interpretada en protocolo  $I^2C$ , que representa la intensidad del odorante con respecto a la posición de percepción de la fuente del robot. La única referencia electrónica real que se ha tenido es el trabajo de fin de carrera desarrollado por Tomás Vásquez Rubio en abril de 2013 [8], en donde se explica que la nariz electrónica empleada ha sido desarrollada en GNB por David Yañez, utiliza protocolo programable  $I^2C$  para las comunicaciones, siendo esta característica muy ventajosa, pues no se tiene sino que programar el protocolo para que pueda introducirse la señal a la Beaglebone, o mejor dicho leer la programación del protocolo por una de las dos entradas  $I^2C$  que posee. Esto habla de la gran capacidad de comunicaciones que posee Beagle.

Como era lógico suponer esta nariz se polariza y devuelve señales digitales de 0 a 5Vdc, que no son aceptables y que podrían producir la destrucción de la Beaglebone, así que

nuevamente se deben acoplar estas señales de 0 a 3.3Vdc. Para ello se utiliza otro C.I. de Texas instruments que esta seriado por PCA9306 que puede aceptar voltajes de entrada desde 1.2Vdc para salidas de 1.8Vdc, 2.5Vdc, 3.3Vdc y 5Vdc; varias alternativas más y la que se ha utilizado, entrada de 5Vdc y salida de 3.3Vdc. La distribución circuital y la conexión desarrollada en altium designer se presentan en la Figura 4.15



**Figura 4.15 Disposición y conexión del PCA9306.**

Como se observa en la Figura 4.15, para la adaptación efectiva de los voltajes requeridos se deben utilizar unas resistencias de pull up, que se calculan de acuerdo a los voltajes de referencia que se necesitan, de acuerdo a la siguiente fórmula:

$$R_{PU} = \frac{V_{DPU} - 0.35 V}{0.015 A}$$

Donde RPU y VDPu son los voltajes y sus correspondientes resistencias en cada lado, es decir en este caso 3.3Vdc y 5Vdc. Los valores obtenidos son 220Ω y 341Ω respectivamente.

La línea EN permite habilitar al dispositivo y las líneas SCL1 y SCL2 sirven para la señal del reloj de sincronía de la nariz y SCA1 y SCA2 para los datos enmascarados en protocolo I<sup>2</sup>C. Estas señales se obtienen mediante un conector molex de 4 pines conectado en las cercanías del circuito integrado.

## 4.10 Sistema de comunicaciones

Para las comunicaciones entre los robots se ha decidido entre las varias alternativas presentadas, comunicación Zigbee, a través de módulos XBee PRO de serie 1, ya que tienen las siguientes ventajas:

- Facilidad de programación y configuración.
- Posibilidad de conexión en red.
- Comunicaciones inalámbricas.
- Fácil integración.
- Conexión sencilla y de muy pocas líneas con Beaglebone a través de UART.

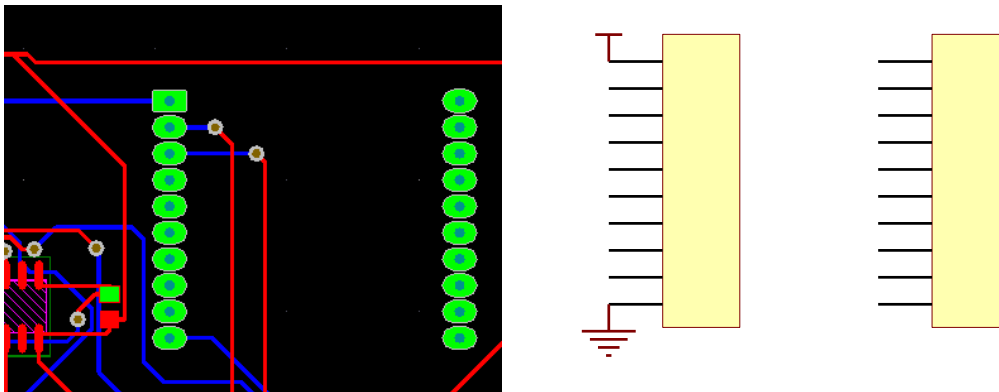
Beaglebone presenta la posibilidad de comunicaciones vía cable con una interfase serial por UART, de hecho, existen cuatro puertos seriales disponibles en diferentes configuraciones, las cuales se establecen en transmisiones seriales con control y sin control de flujo, para este caso se ha utilizado solamente señales de polarización y las líneas de transmisión y recepción de la señal.

XBee PRO series 1 posee una antena para transmitir las señales de información a través de ondas de hasta 2,4 GHz; Según la información encontrada en el sitio oficial de la marca fabricante [9], manifiestan que: Los módulos XBee proveen 2 formas amigables de comunicación: Transmisión serial transparente (modo AT) y el modo API que provee muchas ventajas. Los módulos XBee pueden ser configurados desde el PC utilizando el programa X-CTU o bien desde un microcontrolador. Los XBee pueden comunicarse en arquitecturas punto a punto, punto a multi punto o en una red mesh. La elección del módulo XBee correcto pasa por escoger el tipo de antena (chip, alambre o conector SMA) y la potencia de transmisión (2mW para 300 pies o 60mW para hasta 1 milla). En este caso hemos elegido una antena dipolo de tipo cable, por la altura que presenta el robot.

La serie 1 está basada en el chipset Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el chipset de Ember y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red mesh. Ambos módulos pueden ser utilizados en los modos AT y API.

Los módulos XBee pueden ser usados con sus propios adaptadores XBee Explorer Serial o XBee Explorer USB. Para el trabajo de elementos sean micros o placas de 5Vdc, se necesitará un elemento acoplador, en el caso particular del robot, no fue necesaria ninguna adaptación ya que el header de Beaglebone presente de forma normal una fuente de salida de 3.3Vdc y los pines de I/O presentan y permiten la misma tensión.

Debido a que los módulos XBee tienen una separación de pines de 2mm el fabricante recomienda utilizar una de sus tarjetas adaptadoras, en el robot construido se desarrolló el footprint del conector con las especificaciones obtenidas del fabricante (figura 4.15.), conectándolo directamente con la tarjeta adaptadora. Esta librería la ponemos a disposición de cualquier persona que utilice altium designer. En la Figura 4.16. se observa la disposición y conexión de altium designer y el footprint construido.



**Figura 4.16 Footprint y conexión de XBee en la placa adaptadora.**

Como se observa se utilizó un par de conectores header de 10 pines y en la parte de realización del PCB se cambió el footprint con uno que se construyó expresamente. Y se destaca que se conectan únicamente las señales de polarización de 3.3Vdc y la entrada y salida de datos seriales que serán los que transmita la antena.

## **4.11 Uso de recursos de Beaglebone.**

Como se ha mencionado en innumerables ocasiones anteriormente, Beaglebone posee enorme capacidad de recursos de comunicaciones y gestión de su misma plataforma, y que sería imposible utilizar todo en una sola aplicación, pues el multiplexor interno, es quien designa que forma de operación usa cada pin de entre 6 o 7 diferentes formas. De todas formas, el robot construido utiliza de manera separada cada recurso hardware, pues el número existente de ellos es adecuado a lo dimensionado para cada aplicación, y además no se traslapa de ninguna forma con otro disponible.

En la Figura 4.17 y Figura 4.18 se indican los header de Beaglebone, y se detallan las señales que se utilizan. A continuación se explica cómo se usan cada uno de los pines de cada header. Para ello es necesario que, ante cualquier duda se remita a las tablas 4.2 a 4.7 que se encuentran disponibles en las páginas 52 - 59 y en donde se detallan todas las señales y con qué valor del multiplexor se pueden acceder, es decir para el kernel de

Linux, es el modo de cada pin que se va a utilizar. Esto es importante porque el software debe asegurar que cada pin funcione en el modo adecuado.

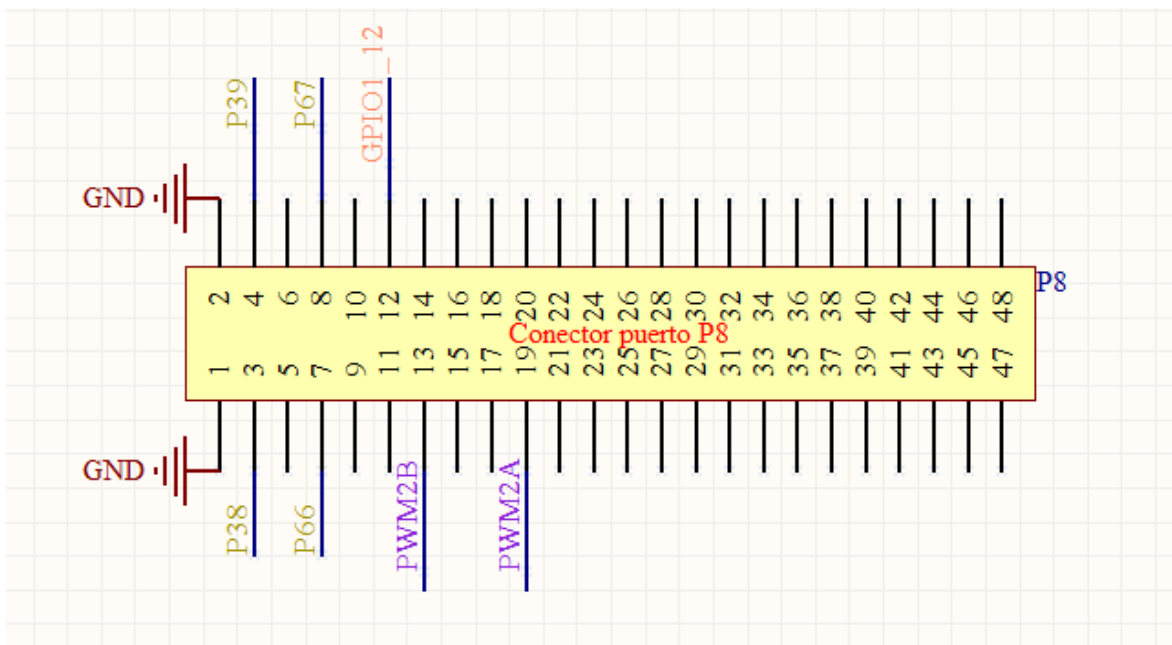


Figura 4.17 Header de Beaglebone puerto P8

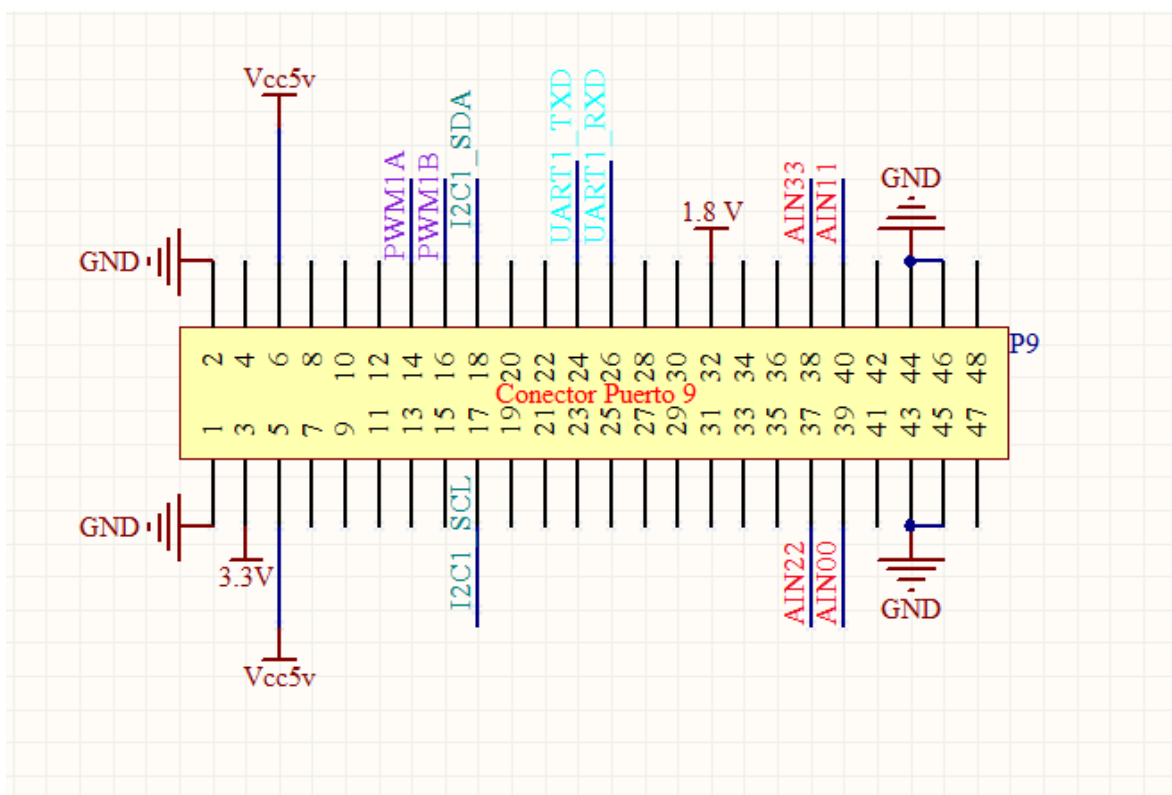


Figura 4.18 Header de Beaglebone puerto P9

Se observa que del header P8 solamente se utilizan algunos pines, a saber los pines 3, 4, 7 y 8 que se usan para la entrada y control de los encoders de cada uno de los motores y se han designado por P38, P39, P66 y P67, por la consideración de que ellos crean los gpio de entrada 38, 39, 66 y 67, que en última instancia serán los que se programan y que vienen desde el buffer de conversión de la tarjeta de los encoders polarizada con 5 voltios, y corresponden a los encoder A y B de cada rueda. Se utilizan en el modo 0.

El pin 12 signado como gpio1\_12 y que será utilizado como entrada digital de 3.3Vdc para el control de sobrecarga e incremento de la temperatura del driver DRV8833, proveniente de la señal fault del driver. Se utilizará en el modo 1.

Los pines 13 y 19 han sido reservados para el uso de las líneas PWM para el control de dirección y velocidad del motor izquierdo, se utilizarán en el modo 6.

Para el puerto 9 se han utilizado muchas señales, en primer lugar el pin 3, corresponde a la fuente de 3.3Vdc que entrega el circuito regulador de voltaje de la placa Beaglebone, a partir de los 5Vdc de alimentación a través de su Jack estándar.

Los pines 5 y 6 tienen una doble funcionalidad. Primero, cuando se conecta la fuente electrónica al Jack de la Beaglebone, estos dos pines proporcionan una toma directa de esta fuente, suministrando una corriente de hasta 1.5A, y en segundo lugar, son pines que aceptan la conexión de una fuente de 5Vdc por medio de unos plugs directos al header, que polariza por completo y da funcionamiento a la placa, en este caso la placa adaptadora entrega la tensión proveniente del regulador a la Beaglebone, dándole de esta forma la autonomía necesaria al robot, hay que recordar que esta placa funciona con la batería portátil.

Los pines 14 y 16 son las salidas PWM para los dos conectores del motor derecho, configurables de idéntica forma que las salidas del header P8 explicadas antes. La entrada 17 corresponde al reloj del protocolo I<sup>2</sup>C proveniente de la nariz y el pin 18 a la señal de datos del puerto I<sup>2</sup>C de la nariz. Son provenientes del conector de la nariz.

Los pines 24 y 26 son los encargados de enviar las señales desde el robot al medio inalámbrico a través del UART y recibir la información proveniente de este sistema de comunicaciones, estas señales están conectadas al módulo XBee.

El pin 32 es la fuente de referencia para los buffer conversores de las entradas analógicas y su señal de habilitación, es una fuente de 1.8 voltios, proveniente del regulador de voltaje de la placa Beaglebone.

Finalmente los pines 37, 38, 39 y 40 son las entradas analógicas que maneja la placa y que provienen de los sensores de obstáculos Sharp, pasando por un buffer conversor, pues son puertos de 1.8Vdc para los ADC de la Beagle, pero que provienen de placas de los sensores SHARP polarizados con 5Vdc. Físicamente se los obtiene de los cuatro conectores molex de tres pines de la placa.

## 4.12 Construcción del Hardware relacionado.

Como ya se ha mencionado en otras ocasiones anteriormente, las placas utilizadas en este proyecto fueron diseñadas en Altium Designer. El resultado del ruteo y la construcción del PCB, se explicará a continuación, así como los inconvenientes y adecuaciones que se tuvieron que realizar a estos elementos.

El PCB se realizó en el taller de circuitos impresos de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, en la máquina fresadora ProtoMat S100, que rutea los caminos de los PCB diseñados en altium designer de hasta una anchura de 0.2mm y los agujeros los taladra hasta de 0.1mm. Para ello se deben disponer los archivos Gerber generados en altium designer, comprobarlos, verificar el estado de las cuchillas, verificar la correspondencia exacta de las dos caras de las placas y proceder. En las Figura 4.19, 4.20 y 4.21 se muestra el resultado final de la placa adaptadora construida, así como los resultados del ruteo y sus implicaciones en cada etapa mencionada en el diseño.

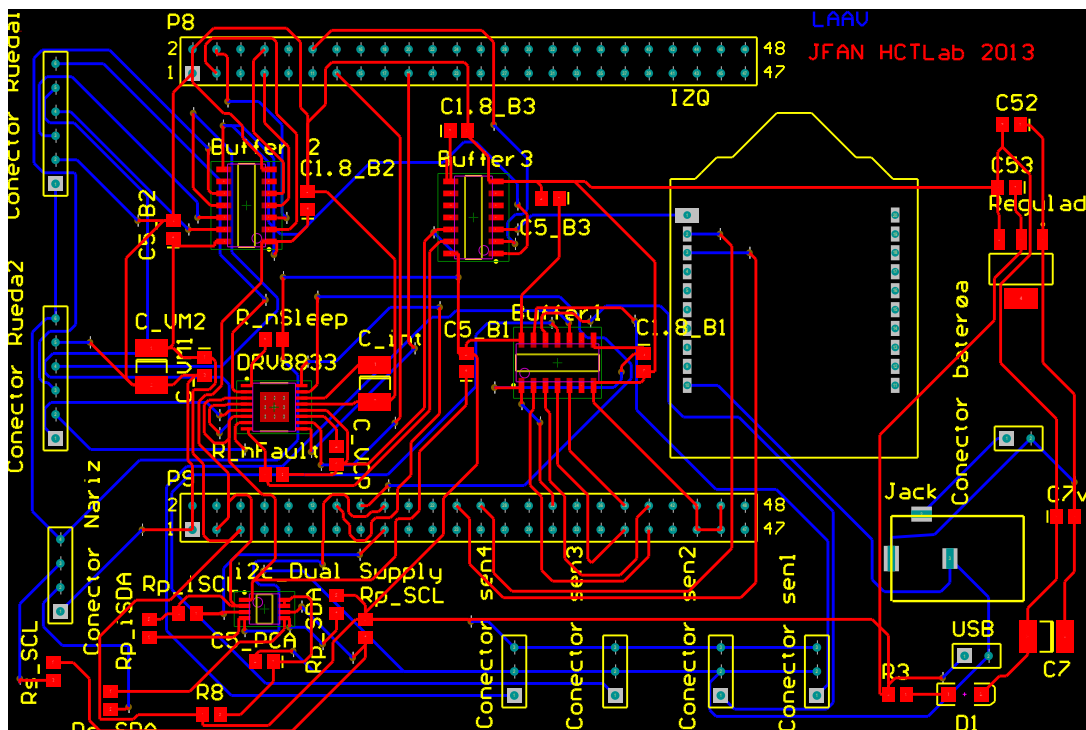
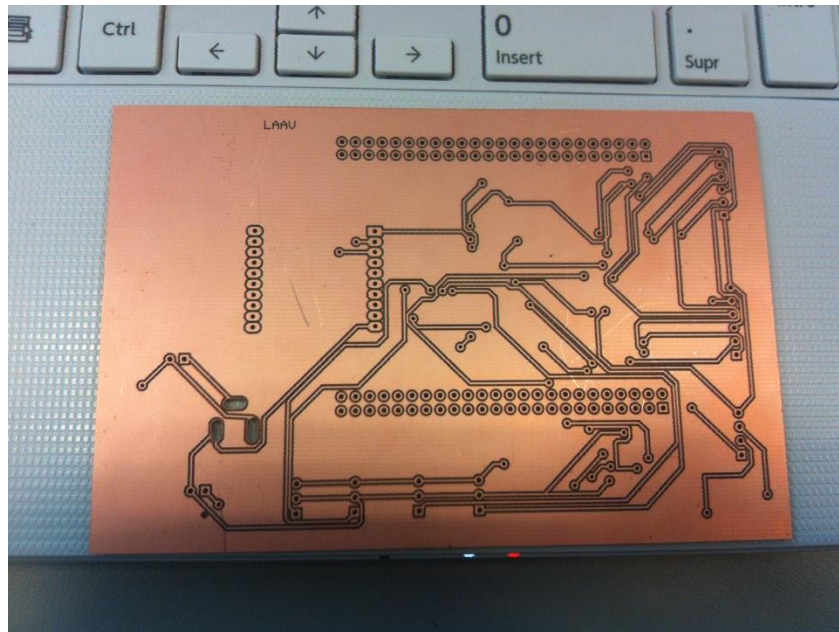
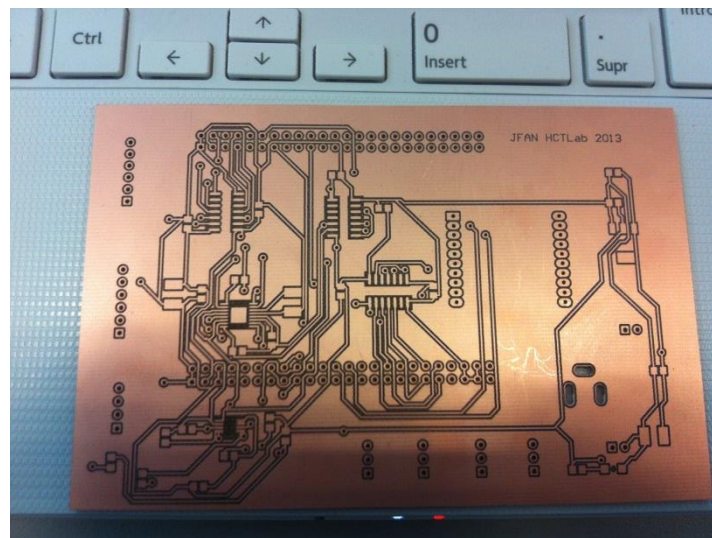


Figura 4.19 Conexiones Ruteadas de la placa adaptadora





**Figura 4.20 Vista posterior de la placa adaptadora**



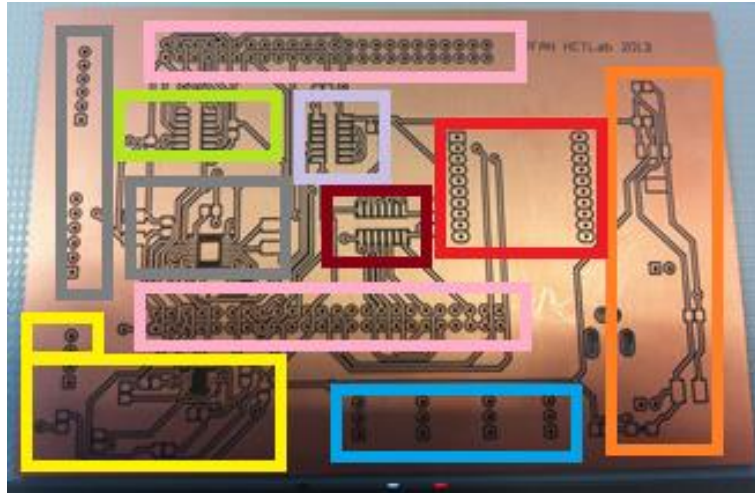
**Figura 4.21 Vista superior de la placa adaptadora**

Se puede verificar que la posición de los elementos conectores y las vías de conexión de los elementos activos y pasivos del PCB coinciden en cierto sentido con los del diagrama que se presentó en la hoja de trabajo de altium designer, la razón es que se quiere presentar de forma intuitiva para cualquier persona que desee investigar su funcionamiento.

Se verifica además que las distancias entre los conectores en donde se colocarán los pines de los header de las Beaglebone, son exactamente del tamaño adecuado, tal como lo había especificado el fabricante, y esto se da para producir la perfecta conexión entre las

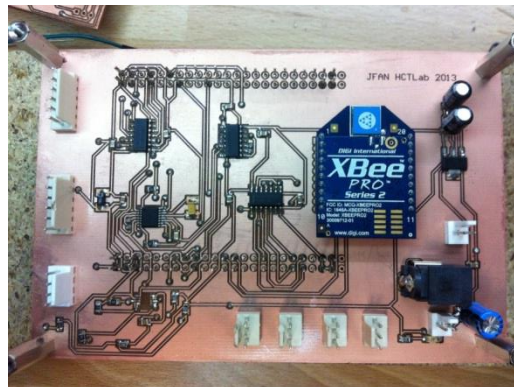
placas, además de que también se observa el conector del módulo XBee con su distancia y espaciamiento entre pines exacto, que se había dicho, se fabricó a medida, desarrollando la librería del footprint.

A continuación en la Figura 4.22 se identifican gráficamente las etapas de conexión de la placa adaptadora.



**Figura 4.22 Secciones de la placa destinadas a bloques funcionales.**

La sección marcada de naranja corresponde a las fuentes de alimentación y regulación de la tensión de consumo. La sección azul, a las entradas de los sensores de obstáculos, la sección amarilla corresponde al conector de la nariz y su driver, la sección gris corresponde al manejador de los motores con sus conectores, y de retorno de las señales de los encoders de cada motor, el rectángulo color limón, corresponde al buffer manejador de los encoders, el gris al buffer manejador de 3.3Vdc, el rojo a la sección de las comunicaciones Zigbee, el morado al buffer de las entradas analógicas y los rosados los conectores de unión con los header de la placa Beaglebone.

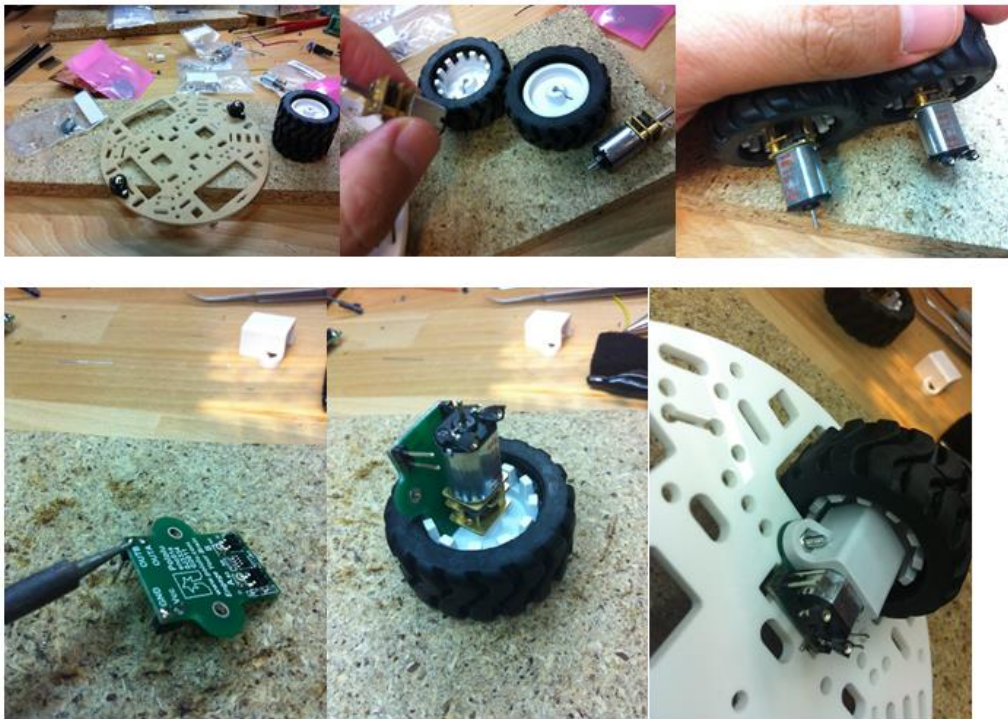


**Figura 4.23 placa adaptadora terminada**

En la Figura 4.23 se observan ya los elementos que se han instalado en la placa para la extensión hacia los sistemas de motricidad del robot, lectura de su posición, detección de obstáculos, sistema olfativo y sistema de comunicaciones, además de su sistema de alimentación global. Refiérase a la Figura 4.21 para comparar con la placa virgen, las superficies destinadas a cada sistema.

#### **4.13 Construcción del chasis y procedimientos de acoplamiento y operación.**

Como se mencionó al inicio de esta sección, para la parte motriz del robot se eligió el conjunto de ruedas, motores y encoders de Pololu, que es una empresa que en gran parte destina sus esfuerzos a la implementación de sistemas robotizados. El conjunto total lo completa el soporte del robot (chasis) que ha sido diseñado por Pololu de tal forma que no se deban hacer adaptaciones mecánicas (soportes de motores, encoders, agujeros de las ruedas) sino más bien se utilicen las existentes, es el soporte de serie 1505 [10], que posibilita realizar la unión de los componentes solamente con tornillos sujetadores, es así que en la figura siguiente se observa el proceso de montaje del chasis.



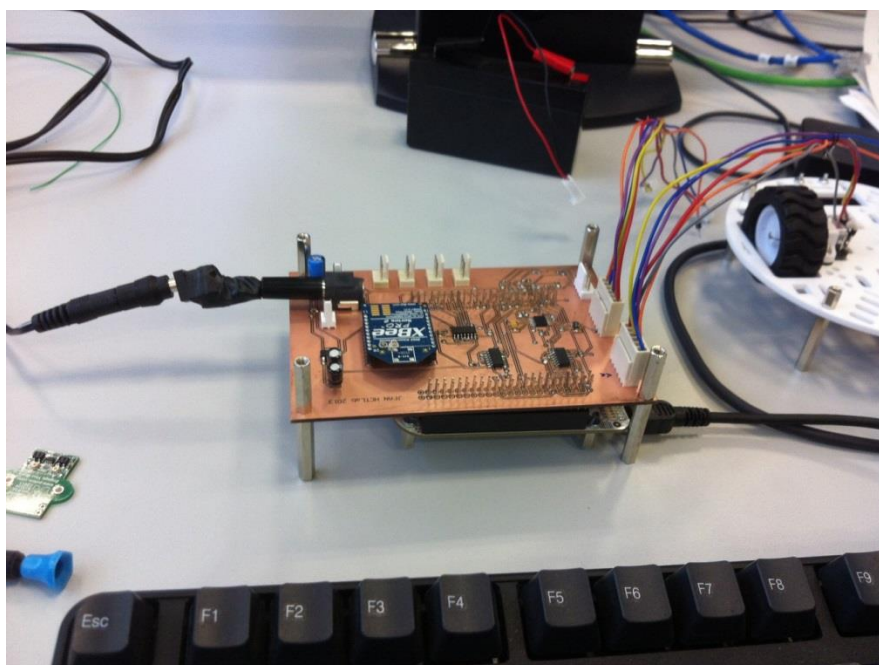
**Figura 4.24 Proceso de montaje del chasis.**





**Figura 4.25 Chasis terminado.**

Posteriormente se acoplan el chasis a su componente electrónico y se procede a las pruebas de funcionamiento y la programación del sistema inteligente.



**Figura 4.26 Integración electrónica y mecánica.**

Los motores pueden funcionar de acuerdo a las configuraciones presentadas en las tablas 4.8. y 4.9.; pero para mayor seguridad y asegurar que los puentes H hagan su trabajo de la forma más eficiente posible se ha utilizado el método de PWM, con la lógica que se detalla en la Tabla 4.10 a continuación:

**Tabla 4.10 Lógica de funcionamiento de los motores**

<b>xIN1</b>	<b>xIN2</b>	<b>FUNCTION</b>
PWM	0	Movimiento directo
0	PWM	Movimiento reverso

Pudiendo controlarse la velocidad de las ruedas variando la relación de trabajo de las señales PWM por software, pero para este caso no se considerarán los valores completos de la variación de relaciones de la señal PWM porque los niveles de voltaje medio resultante de un gran porcentaje de ellos no tiene el suficiente valor para activar los puentes H, es así que solamente se utilizan las variaciones desde 60% al 99%, pudiéndose controlar la frecuencia de la señal en un amplio rango, en este caso se ha establecido entre los 1KHz hasta 1 MHz, que hace que estos rangos antes mencionados tomen saltos más pequeños y su diferencia de velocidad sea imperceptible.

De esta forma se controla los dos motores, haciendo posible el movimiento para adelante, en reversa, giro a la izquierda y giro a la derecha, controlando el radio de giro, desde el centro al extremo del robot o desde extremo a extremo, es decir giros más rápidos o más lentos.

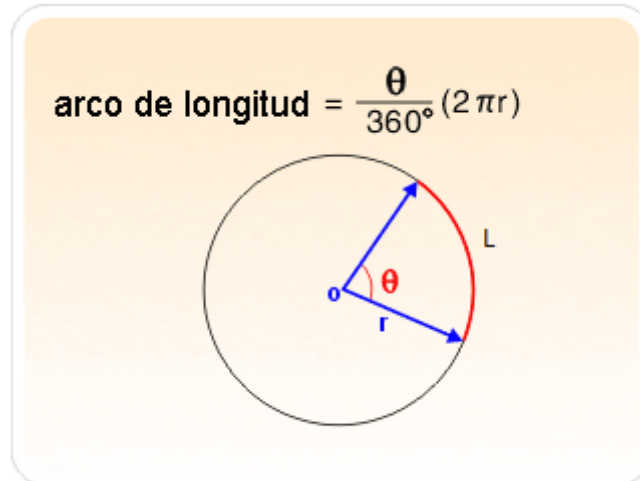
El control del desplazamiento del robot se lo puede hacer de dos formas, por tiempo o por posición, es decir distancia, en el primer caso se sincronizan los movimientos a las acciones internas de la CPU y en el segundo caso los movimientos dependen de los valores entregados por los encoders de cada rueda. La distancia de movimiento lineal del robot depende del radio de la rueda, pues el desplazamiento lineal de la misma, corresponde a la longitud del arco descrito para cada ángulo de giro, de acuerdo a las siguientes constantes y ecuaciones:

Utilizando la fórmula trigonométrica, la longitud de arco de una circunferencia de radio  $r$  está definida por:

$$L = \frac{2.\pi.r.\theta}{360^{\circ}}$$

**Ec(4.1)**

De acuerdo a la Figura 4.27



**Figura 4.27 Longitud de arco de una circunferencia**

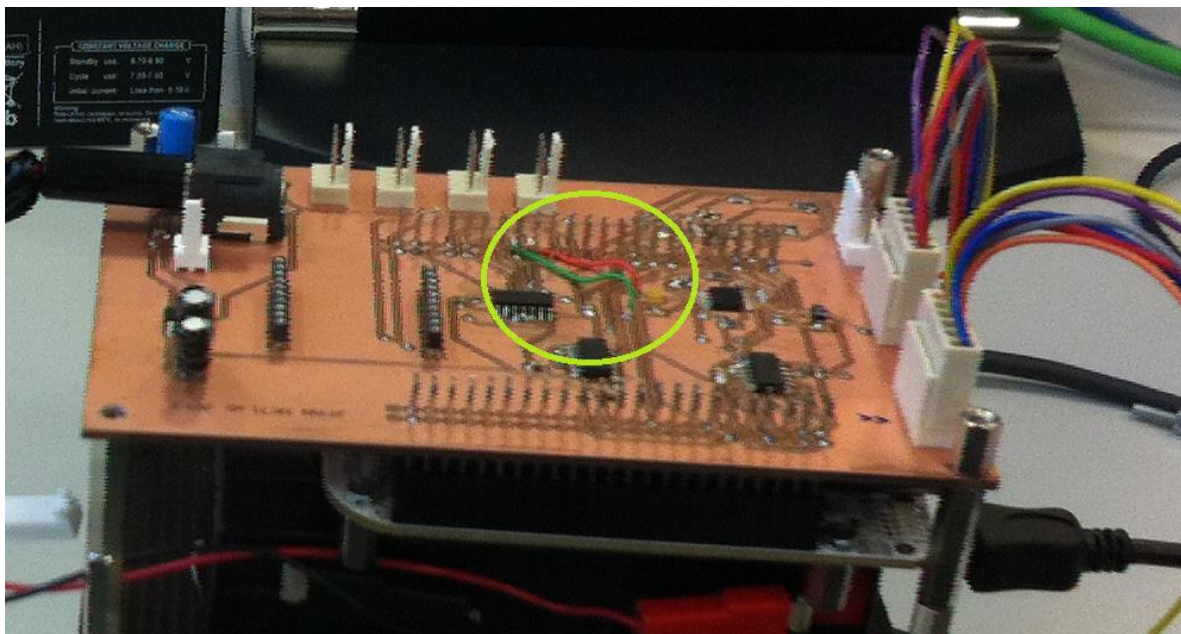
De acuerdo a las especificaciones de las ruedas Pololu es de 2,12cm, lo que resulta para una vuelta completa es decir un ángulo de 360° una distancia de 13,32cm por vuelta. Como en el interior de la rueda existen 12 dientes que leen los encoders como pulso alto cuando detecta un diente y pulso bajo cuando es una ranura, significa que para cada pulso recorre una distancia aproximada de 1,11cm, que será la base del cálculo de la posición del robot por cada movimiento que realice.

El software considera el conteo de pulsos para el desplazamiento del motor y los puede hacer de una forma absoluta o una forma comparativa, pues existen dos encoders por cada motor y dos motores, es decir, se puede tomar en cuenta un solo encoder de un solo motor y definir su desplazamiento, o un solo encoder de cada motor y definir una valor medio de los pulsos producidos (pues las ruedas en sus dientes y ranuras no necesariamente deben estar completamente alineadas) y así calcular el desplazamiento; o finalmente para mayor exactitud, leer los dos encoders de cada motor para sus cuatro estados (diente-diente; diente-ranura; ranura-diente; ranura-ranura) y sacará de igual manera que el caso anterior el promedio de los estados contados y medirá de esta forma la distancia recorrida.

Los encoders también pueden predecir la dirección de movimiento, pero este caso en los programas de este robot no ha sido tomado en cuenta, porque se le ordena tener tal o cual movimiento en específico y se sabe qué movimiento es de forma tácita, porque los movimientos son controlados de forma que el robot “busque” la fuente de olor con los movimientos disponibles.

En el manejo de los motores se tuvo un problema serio, que originó se manipule la integridad de la placa adaptadora, pues originalmente como se puede verificar en el

manual Beaglebone SMR, en la páginas: 26 y 54 a 66, indica que la plataforma Beaglebone posee 8 salidas de PWM configurables y que están disponibles en los pines de los headers correspondientes, llamados EHRPWMXX, donde XX corresponde a 0A, 0B, 1A, 1B, 2A y 2B, y los especiales ecap0 y ecap1. Esto hizo que lamentablemente en el diseño de altium designer se considere el uso de EHRPWM1A, 1B, 2A y 2B desde los pines P8.13 y P8.19 para 2A y 2B y P9.14 y P9.16 para 1A y 1B, por razones de no usar pines duales y uso ponderado de los dos headers, pero en el kernel de Linux en la versión de Armstrong 3.5-2 solamente existían 0A, 0B, 1A y 1B. Luego de investigar profundamente la solución y probar unos programas de habilitación de los puertos, se llegó a la conclusión de que era imposible hacerlo, entonces se tuvo que cortar las pistas de la placa que llevaban a esas conexiones y cablear dos conductores a las salidas de EHRPWM0A y 0B correspondientes a los pines P9.31 y P9.29 respectivamente. Como se puede apreciar en la Figura 4.28.



**Figura 4.28 Modificación de la placa adaptadora.**

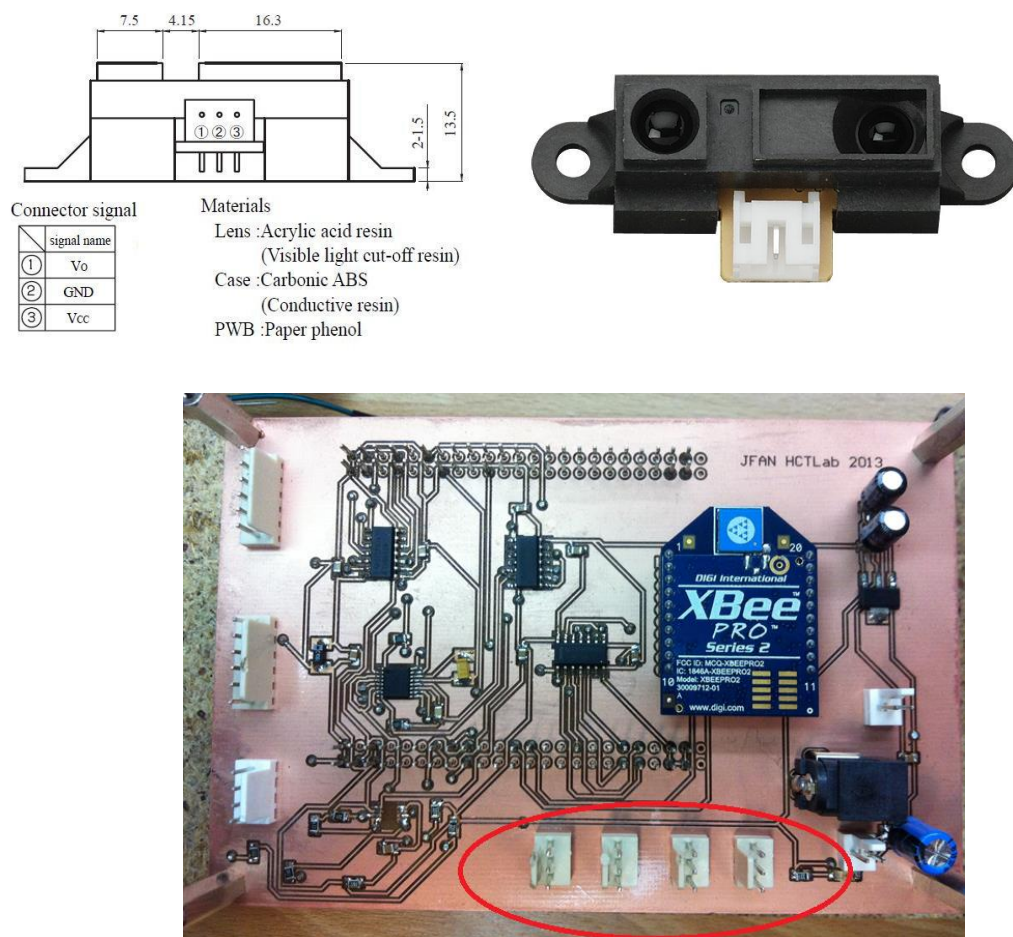
Con esta modificación se consiguió manejar el motor 2 (motor izquierdo) con los PWM antes mencionados.

#### **4.14 Hardware destinado a los sensores de obstáculos.**

La placa adaptadora está dispuesta de forma que se puedan conectar los sensores de obstáculos, pero el hecho de que el trabajo colaborativo exija algunas otras aplicaciones hizo que se dejara de lado tanto la conexión de los sensores análogos de distancia como el software relacionado a su medición e incorporación en el programa principal de operación del robot. Esto no quiere decir que el hardware disponible no esté disponible, al contrario,



lo único que se debería hacer es conectar los sensores a los receptáculos ya conectados en la placa, que son como lo había mencionado en la figura 4.23. los conectores molex de tres pines colocados en su parte inferior. La placa está diseñada originalmente para aceptar sensores de distancia SHARP modelo GP2Y0A21YK0F, que son los que existían en el laboratorio HCTLab a propósito de otros trabajos de fin de carrera que se realizaron en él; pero obviamente no solamente se pueden conectar este tipo de sensores. Más bien podrá aceptarse cualquier sensor que se polarice con 5Vdc y mantenga una respuesta analógica en ese mismo rango o en rangos inferiores, ya que los conectores antes mencionados proporcionan tensión de 5Vdc y su correspondiente tierra y reciben la respuesta referenciada a esta misma tierra, la misma que está conectada a través de un buffer a las entradas analógicas de la Beaglebone correspondientes a los pines del header P9: 37, 38, 39 y 40. Nombradas en el header como ain0, ain1, ain2 y ain3; pero dentro del kernel de Linux como ain1, ain2, ain3 y ain4. Los sensores para los que se implementó el hardware mencionado y la placa correspondiente se indican en la Figura 4.29



**Figura 4.29 Sensores SHARP y conectores en la placa adaptadora, tomado del datasheet del fabricante.**



Se vuelve a repetir que el software no se ha implementado, pero la función realmente no será muy compleja y es parecida a una función que se implementó más adelante para la operación de las entradas análogas.

#### **4.15 Integración de la nariz electrónica.**

Dentro de los objetivos primordiales de este trabajo, consta la integración de una nariz electrónica inteligente para que el robot detecte a través de ella la fuente de odorante, la localice y pueda colaborar con los demás entes, enviando la posición de ésta a través de una red. Es decir era el elemento central del proyecto.

Las narices electrónicas han sido desarrolladas en el grupo de investigación GNB de la EPS, el que iba a colaborar con HCTLab precisamente proporcionando la nariz para que se pueda desarrollar el proyecto. Lamentablemente la implementación de este dispositivo no estaba inmediatamente disponible y se empezó el mismo considerando la forma en que las demás habían sido construidas.

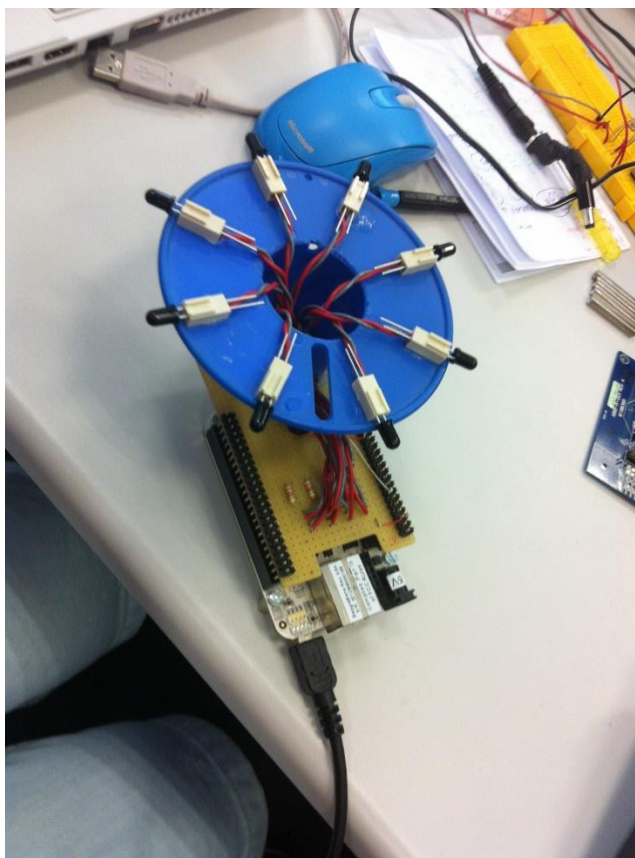
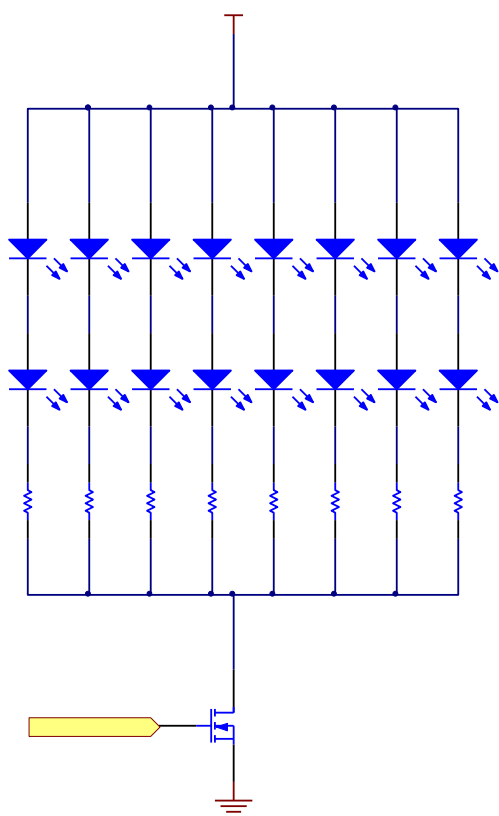
Por la investigación se llegó al trabajo de fin de carrera desarrollado por Tomás Vásquez Rubio [8] que acopló una nariz a un robot oruga mediante un Arduino, esta nariz detectaba la fuente de un odorante específico y convertía la información de concentración del odorante en una señal digital que luego era formateada en protocolo I<sup>2</sup>C y enviada en este caso al Arduino. Con esta consideración, a pesar de pensar particularmente que no era la mejor forma que se pueda recibir la información de olor, pues hubiera sido mejor recibirla como voltaje análogo proporcional, para poder implementar algoritmos de control realimentado. Se diseñó el hardware correspondiente que podía aceptar esa información, es decir el conjunto de conector, detector y manejador que esta implementado en la placa adaptadora y se menciona en la sección 4.10 de este trabajo y se verifica en la Figura 4.15.

Es una lástima pero esto no se pudo implementar porque la nariz no estuvo disponible nunca, entonces la única alternativa era simular el comportamiento de una fuente de olor y de una nariz detectora que fuera implementada en el robot. Para esto se consultó con los miembros del grupo GNB, quienes manifestaron que la “pluma” de dispersión característica de los olores es muy similar a la de la luz infrarroja, entonces se decidió montar una plataforma que generase esa luz de modo omnidireccional y colocar en el robot uno o varios detectores de la luz infrarroja que pudieran dar información proporcional a la intensidad luminosa de la fuente, cuanto más se acercara o alejase el robot de ella.

La plataforma montada tiene dos partes: La simulación de la fuente de olor y la simulación de la nariz artificial.

El primer caso se ha desarrollado una nueva plataforma en una nueva Beaglebone, para comprobar también la versatilidad de la misma. En esta plataforma se ha construido un sistema electrónica que varía la cantidad de voltaje medio que se les entrega a 8 leds infrarrojos de alta potencia con una señal de regulación de dicha variación de la potencia media. Para ello se ha utilizado como entrada una de las entradas análogas de la Beaglebone, la ain0, o ain1 en el kernel, esta señal es obtenida de un divisor de voltaje que ejecuta un potenciómetro de 10 K $\Omega$ . Este voltaje que tiene una variación de 1.8 Vdc, se introduce en dicho pin, y es traducido a una palabra digital de 12 bits que varía en 4096 valores (0 – 4095). Para proteger a la entrada de la Beagle se ha puesto una resistencia en serie que limita la corriente cuando el potenciómetro este a cero, es por eso que no se consideran los valores inferiores de tensión y se manejan a la entradas solamente variaciones de 0,8 Vdc a 1.8 Vdc. Que son suficientes para manejar toda la intensidad que emiten los leds. Esta palabra digital es convertida en forma lineal a su equivalente en “duty percent” de la señal PWM y ese valor se consigna en el archivo de salida del EHRPWM1 de la Beagle, variando así el hardware, que consiste en la alimentación de dichos pulsos a la base de un transistor MOSFET en corte y saturación que cierra el interruptor para dejar pasar la corriente de la fuente de alimentación, que inicialmente fue la Beaglebone, con una fuente electrónica interna, que puede abastecer hasta 1500mA, Pero como los leds pueden soportar hasta 1000mA cada uno, por los cuatro ramales se puede hacer pasar los 4000 mA alimentando de la potencia máxima a los leds y mejorando de esta forma la distancia a la que se puede detectar la fuente de olor. Se puede utilizar una fuente externa que proporcione la corriente mencionada y una adaptación adecuada de las tierras entre esta fuente y la placa Beaglebone.

Esto se observa en el siguiente diagrama electrónico y la imagen de la plataforma construida en la Beaglebone de la Figura 4.30



**Figura 4.30 Simulador de la fuente de odorante**

Los Leds utilizados originalmente fueron los OP293 que poseen una luz radiante de 890 nm, y que puede soportar una corriente al aire libre de 100mA, e irradian su luz con un ángulo de 60°, por lo que fueron necesarios 8 leds para que se cubriera de luz infrarroja toda la circunferencia y se irradian en forma directa y hasta un ángulo constante de 60°.

Se observa que se utilizó la técnica del Wire wrapping, ya que era una placa demasiado simple como para rutearla en un PCB, y además no es parte de la plataforma robótica.

Para los sensores, es decir la nariz electrónica montada en el robot, se montaron 4 sensores de idénticas características: un fotodiodo polarizado inversamente, en serie con una resistencia de 100 K $\Omega$  sobre la cual medimos el voltaje que cae cuando el diodo conduce, es decir cuando tiene la influencia de la luz de longitud de onda y potencias adecuadas, introduciéndose este voltaje en una de las 8 entradas analógicas de la placa Beaglebone.

Para polarizar este circuito es necesario nuevamente recordar que las entradas análogas de la placa Beaglebone solo aceptan voltajes de hasta 1.8 Vdc, es por esto que cada

“sensor” se ha polarizado con la fuente fija de salida de 1.8 Vdc que se obtiene en el pin P9.32 y su correspondiente tierra P9.34.

Estas señales están ingresadas a las 4 entradas analógicas restantes de la placa Beaglebone, que son ain4, ain5, ain6 y ain7, correspondientes a los pines P9.36, P9.38, P9.33 y P9.35, como palabra digital de entrada con un valor de 0 a 4095.

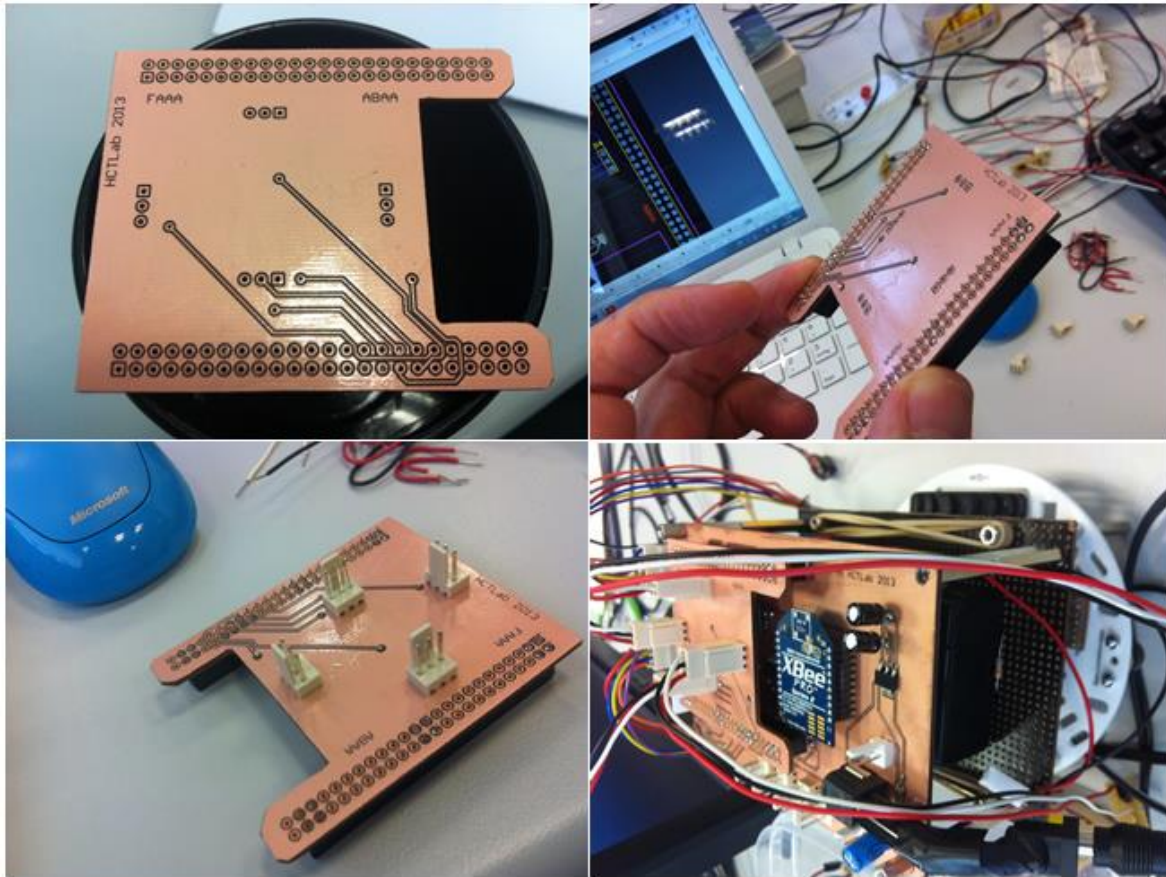
Este valor de voltaje realmente se traduce en la distancia a la cual se encuentra la fuente de luz del sensor y es lo que ha posibilitado realizar el algoritmo de búsqueda.

En este caso el algoritmo de búsqueda y el trabajo colaborativo diseñado en instancias iniciales se modificaron de forma radical, ya que se han colocado cuatro sensores de luz infrarroja en el robot (uno a cada extremo), al contrario de lo que se iba a realizar con la nariz que tenía una sola dirección de flujo, y que hacía que exista la necesidad de buscar esa dirección y después comprobar la intensidad. Acá en cambio, se ha desarrollado un algoritmo que descubra qué sensor tiene mayor intensidad y el robot gire en forma ortogonal y lo encuentre. Más adelante se hablará de este algoritmo.

Para la implementación de este sistema se debió realizar otra placa en PCB, porque el hecho de utilizar los headers implicaba tomar cables soldados o enrollados desde éstos dándole en primer lugar un enorme mal gusto estético al trabajo, y en segundo lugar incomodidad al realizar las conexiones. Afortunadamente en el diseño original se previó que la placa adaptadora considerase en ambos extremos los plugs para conexión, tanto de la Beaglebone en la parte inferior de la misma, como de cualquier otro elemento necesario en la parte superior. Esto se lo realizó porque como se mencionó en las características de la placa en la sección 4.2, Beaglebone ha diseñado y construido los “Capes” de expansión y la intención era de a futuro dotarle de una pantalla LCD al robot para poder visualizar órdenes recibidas y posiciones de los demás agentes, así como conexiones con redes wifi de Ethernet, y conexión a vehículos de combustión interna usando redes CANBUS.

Es así que se diseñó en altium designer y construyó la nueva placa y se le proporcionó entradas de conexión para cada sensor mediante conectores molex de tres pines, que proporcionan 1.8Vdc de polarización y con la misma referencia reciben la tensión en la resistencia.

La Figura 4.31 presenta la placa construida y su acoplamiento en el robot.



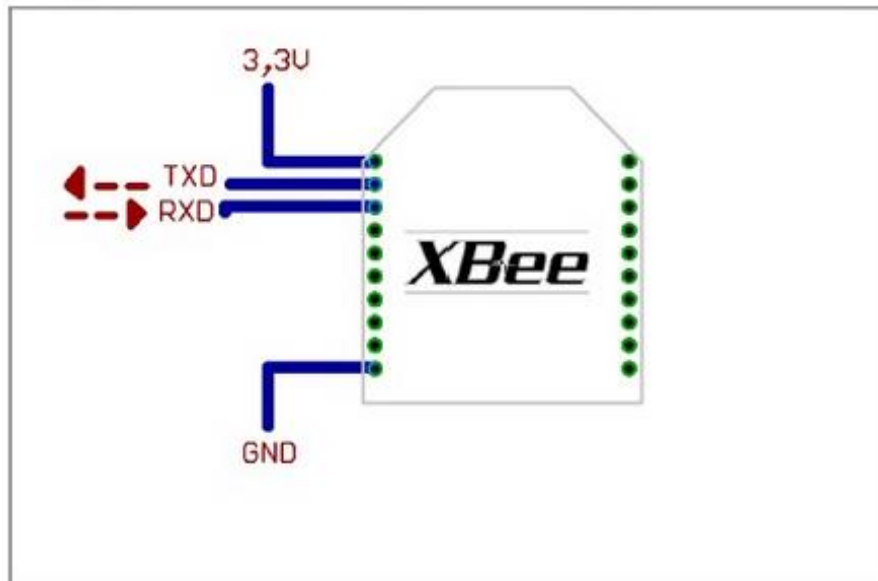
**Figura 4.31** Proceso de montaje de la placa “nariz”

#### **4.16 Incorporación del sistema de comunicaciones.**

Se manifestó en la sección 4.11, y por todas las justificaciones ahí presentadas, que se eligió el sistema XBee para comunicaciones seriales, desde la placa e inalámbricas desde el robot y entre agentes. En la placa madre se puede observar la conexión de este elemento.

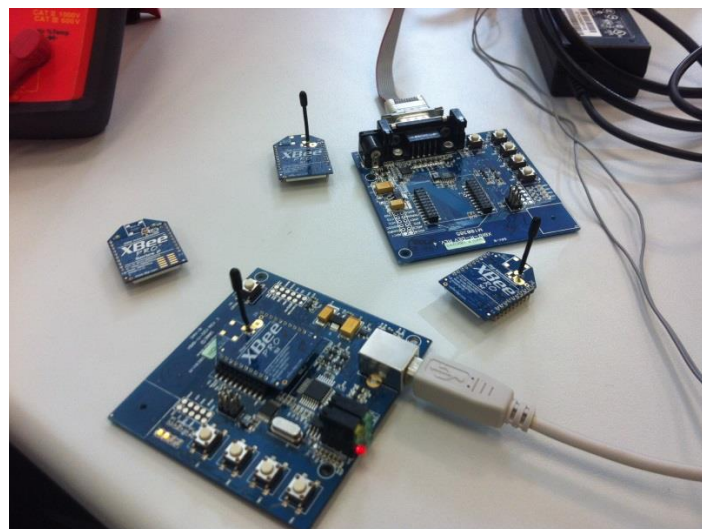
Se pueden conectar módulos de diversos tipos y diferentes especificaciones en la plataforma, ya que XBee permite muchas alternativas, todas ellas trabajando en una banda de hasta 2,4 GHz.

En este caso se utiliza el módulo XBP24-AWI-001 con antena tipo dipolo de cable flexible. Y las conexiones mínimas para la transmisión, que son polarización de 3.3Vdc y tierra y señal de entrada y salida serial, tal como se indica en la Figura 4.32



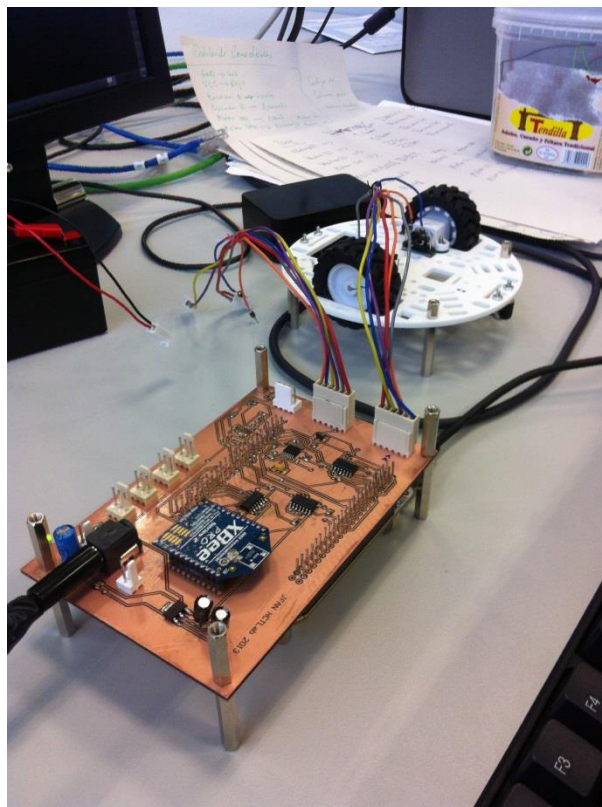
**Figura 4.32 Conexiones mínimas de utilización de un Módulos XBee. Obtenida en el enlace: <http://es.scribd.com/doc/58980339/Teoria-y-Programacion-Modulos-XBEE>.**

Los módulos se configuran aparte a través de interfaces seriales y USB adquiridas para el efecto (Figura 4.33) y posteriormente se prueban en ellas y conectan en la plataforma particular suya (es decir de la misma marca) o expresamente construidas para ello, como microcontroladores o como en este caso sistemas embebidos. La Figura 4.34. muestra esta conexión.



**Figura 4.33 Interfaces seriales y USB para programación de XBee.**





**Figura 4.34 Conexión XBee a la placa adaptadora. (plataforma personalizada)**





## **CAPITULO V**

### **IMPLEMENTACIÓN DEL SISTEMA DE DETECCIÓN DE OLOR.**



## 5 Implementación del Sistema de detección de Olor.

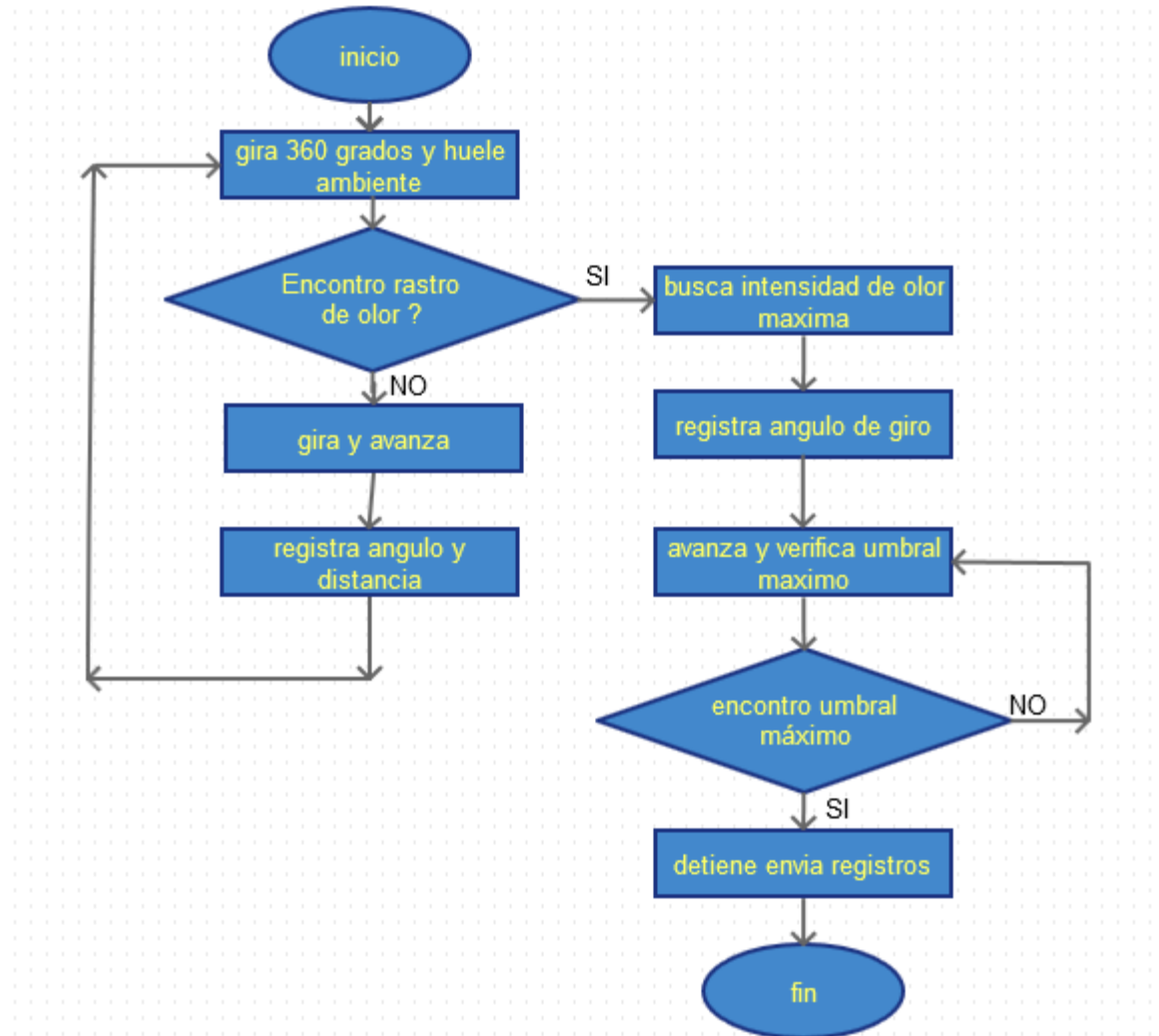
Como se explicó anteriormente, la inconveniencia de no poder contar con las narices electrónicas reales llevó no solamente a re-plantear el hardware que estaba no solo planificado sino construido; hay que recordar que la placa adaptadora tiene instalados los conectores de la nariz, los CI de drivers que manejaban las señales de polarización, reloj y datos del protocolo I<sup>2</sup>C y la conexión con las señales I<sup>2</sup>C del header Beaglebone; sino también a re-planificar la estrategia de detección, localización y colaboración que tendría la plataforma, y así es como ha pasado. Afortunadamente no todo está perdido, la plataforma construida está materialmente lista para en algún trabajo futuro implementar la adaptación hardware de la nariz inteligente, así como de muchas más capas de aplicación que se mencionarán más adelante en la sección de los trabajos futuros; además de que personalmente he planteado a la universidad asociada la continuación de estudios sobre el tema que se ha desarrollado, es decir se puede colaborar para plasmar en realidad los trabajos futuros planteados en forma colaborativa con HCTLab.

A continuación se explican las implicaciones que tuvieron a efecto de no contar con las verdaderas narices:

### 5.1 En la estrategia de búsqueda.

La nariz electrónica que se iba a montar era en la plataforma, el único sensor de su tipo y funcionaba de forma bastante direccional, de tal manera que tenía inclusive un ventilador conectado de forma de extractor de olor hacia ella, que hacía súper direccional el flujo de aire que contenía eventualmente el componente de odorante. De esta forma se diseñó un algoritmo que haga que el robot se mueva “aleatoriamente” por una superficie definida, a partir de una posición conocida que era el punto de coordenadas (0,0); y en intervalos de tiempo definidos por las pruebas, se detenía y giraba 360° hasta encontrar “rastros” del odorante; este trabajo lo hacía continuamente y en forma idealmente infinita si no encuentra “novedad” en el ambiente. De encontrar en uno de sus movimientos la fuente de olor con una intensidad mayor que un umbral establecido, disminuía la velocidad de rotación al mismo tiempo que iba comparando y registrando los valores dados de intensidad de olor, hasta que encontraba el valor máximo. En ese punto se detenía y registraba el ángulo de giro. Inmediatamente avanza hasta encontrar el umbral máximo de la fuente de olor y se detiene, entrega los datos por el sistema de comunicaciones que ya tiene también registrado los datos de todos los movimientos del periodo, puede ser una hora, un día, etc. Eso puede residir en una base de datos en el sistema supervisor o en

un sistema datalogger conectado al mismo robot a través del puerto USB. Este algoritmo se diseñó de acuerdo al siguiente diagrama de flujo (Figura 5.1)



**Figura 5.1 Diagrama de flujo de la estrategia de detección del olor con una nariz electrónica.**

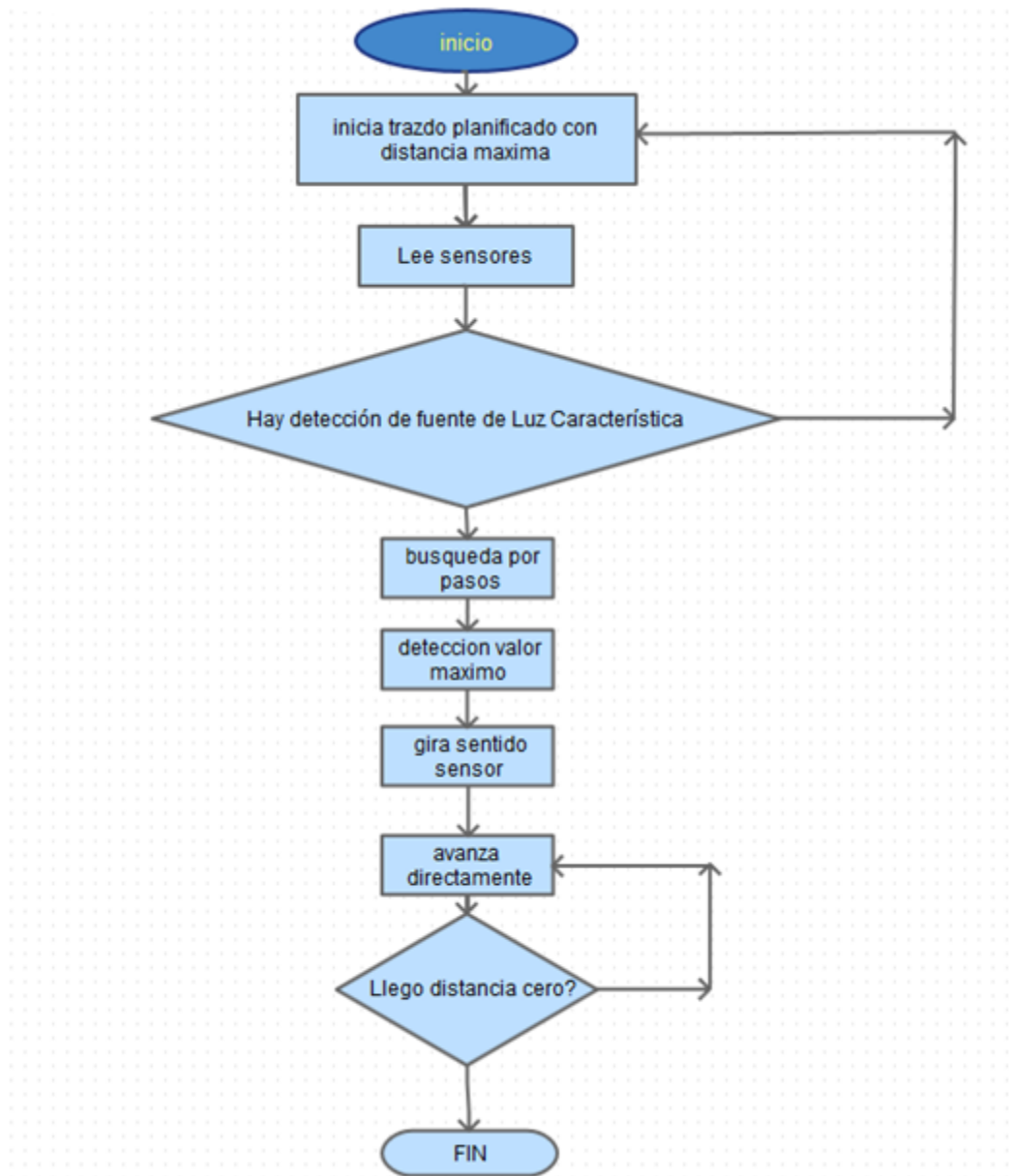
Para la nueva plataforma implementada, es decir para la “simulación” de la nariz se emplea un procedimiento diferente, pues no existe un solo sensor, sino cuatro sensores colocados estratégicamente en las paredes del robot, esto por la escasa distancia que discriminan los sensores, con uno solo hubiera sido materialmente imposible detectar la fuente de luz del emisor y así emplear la misma estrategia diseñada originalmente, entonces se la cambia en concepto, pero en operación se hacen modificaciones radicales a los movimientos del robot en su conjunto.

La estrategia cambia en la forma en la que se busca la fuente de olor. En primer lugar los movimientos iniciales e infinitos del robot ya no pueden ser aleatorios, tienen que tener

un trazado planificado, entonces, para poder monitorear la misma área, se hará que el robot “barra” toda esa superficie a modo de una espiral “cuadrada”, es decir primero su perímetro completamente y cuando llegue a su punto inicial reduzca en forma iterativa y automática su distancia a recorrer. Y lo vuelva a describir, en este caso se estará recorriendo un cuadrado de menos área, definida por el programa y que depende mucho de la distancia en que sean capaces de leer los sensores de luz el umbral mínimo necesario para considerarse como fuente de olor, además que ésta puede presentarse en cualquier instante en cualquier sitio de su área. Esto lo hará, hasta estar en el centro del cuadrado, entonces gira y regresa al punto (0,0) y vuelve a repetirlo, hasta que se agote la batería si no detecta una fuente de olor. La ventaja de esta estrategia es que solo se registra el valor de distancia recorrida de un eje, porque se conoce tácitamente los valores del otro. Situación que no la resolverá el robot, sino el sistema de supervisión que necesita para la colaboración.

En caso de que encuentre la fuente de olor procede de la siguiente forma: mientras se mueve monitorea continuamente los 4 sensores y verifica si alguno de ellos o dos de ellos (cosa que puede suceder frecuentemente) han superado el umbral mínimo de detección. Entonces para y a continuación sigue avanzando en su misma dirección con una velocidad más lenta (por pasos) leyendo únicamente y de forma continua los sensores involucrados en la detección, aquí uno de ellos debe reducir su medición y el otro aumentarla, se monitorea a este sensor hasta que llegue a su valor máximo, en ese momento se gira 90° en el sentido del sensor y se mueve a velocidad normal hasta encontrar el umbral máximo solo con el sensor principal (que viene a ser el frontal), entonces se detiene y envía el valor del movimiento y se determina las coordenadas, en el sistema de supervisión.

Esto se muestra en el diagrama de flujo presentado en la Figura 5.2



**Figura 5.2 Diagrama de flujo de la detección de la fuente de luz.**

Para darle la funcionalidad requerida al robot, y hacer que realice de forma normal los movimientos y acciones descritas, se han implementado funciones de lenguaje C++ e integrado en un programa principal que se presenta en los anexos a este trabajo.

Las funciones escritas han sido determinadas por el recurso que se desee utilizar y hacer uso de todos los que estén disponibles para ello, que como se ha visto en la placa Beaglebone son muy extensos.

El sistemas de simulación de los odorantes tiene limitaciones de distancia, lo que hizo que su estrategia sea modificada hasta poder cubrir con ella un determinado espacio en donde reside el robot, es así que se considera que deben existir una especie de celdas cuadradas de celular cada una de ellas con una agente, en donde se detecte la presencia de los odorantes. El software que se ha desarrollado se ajusta a ese algoritmo y se describe en las funciones creadas para el efecto.

En primer lugar se ha utilizado directamente el kernel de Linux a través de una terminal y se ha comprobado el funcionamiento de las diferentes instancias digitales y analógicas y probado cada una de ellas, y a partir de ahí se investiga la forma de ejecutarlas en un lenguaje de programación. Como se mencionó anteriormente las alternativas para esto son: Java Script, Phyton, y Lenguaje C; En la sección 4.4 se explicaron las razones por las cuales se eligió realizarlo en este lenguaje. Las funciones escritas se basan en las siguientes líneas de programación del kernel de Linux y operarlas directamente en la Beaglebone:

- Para manejo de una entrada digital por el P9.25 llamado mcasp0\_ahclkx (ver tabla 1.3):
  - # cd sys/kernel/debug/omap\_mux
  - # echo 7 > mcasp0\_ahclkx
  - # cd sys/class/gpio
  - # echo 117 > export
  - # cd gpio117
  - # cat direction
  - # cat value
- Para manejo de una salida digital con led de usuario y led de gpio desde el pin P8.3 llamado gpmc\_ad6.
  - En led de usuario
  - # cd sys/class/leds/Beaglebone::usr2
  - # echo 1 > brightness
  - En led de gpio
  - # cd sys/kernel/debug/omap\_mux
  - # cat gpmc\_ad6
  - # echo 7 > gpmc\_ad6
  - # cd sys/class/gpio
  - # echo 38 > export
  - # cd /sys/class/gpio/gpio38
  - # cat direction
  - # echo out > direction

- # cat direction
- # echo 1 > value
- # echo 0 > value
- Para manejo de una entrada analógica desde el pin P9.39 llamada ain0.
  - # cd sys/devices/platform/tsc
  - # cat ain1
- Para manejo de una salida analógica mediante una señal de modulación por ancho de pulso PWM por el pin P9.14, llamado gpmc\_a2 que maneja el ehrpwm.1:0.
  - # cd sys/kernel/debug/omap\_mux
  - # echo 6 > gpmc\_a2
  - # cd sys/class/pwm/ehrpwm.1:0
  - # echo 1 > request
  - # echo 1 > period\_freq
  - # echo 50 > duty\_percent
  - # echo 1 > run

En todos los casos se puede verificar que la instrucción “echo” lleva al recurso correspondiente un valor determinado (después del signo >), y la instrucción “cat” verifica el valor que tiene cada recurso. Por otro lado también se observa que en los subdirectorios del kernel se escoge la forma de operación del pin que se desee o necesite, es decir llevándolo a éste al modo de operación que determine el multiplexor, de entre los cuantos modos disponibles (muchos pines contienen funcionalidad hasta en los 7 modos pero algunos de ellos solamente la tienen en 1, como el caso de las entradas análogas).

Otro aspecto importante que se debe manifestar, es que los puertos gpio (65 en total) pueden ser configurados como entrada o salida y que cada uno de ellos tiene un valor que es el valor operativo en el kernel de Linux Ubuntu y se lo determina por la siguiente ecuación:

Tomando en cuenta la definición GPIO<sub>A\_B</sub>

$$N_{PUERTO} = 32 * A + B$$

Ejemplo el GPIO1\_6 corresponde al directorio del kernel de Ubuntu gpio38.

Finalmente hay que considerar muy seriamente los niveles de tensión manejados por Beaglebone: todos los GPIO, señales de I/O, McASP0, SPI1, I<sup>2</sup>C, GPIO(65), LCD, GPMC, MMC1, MMC2; manejan señales de 3.3 Vdc y las entradas análogas 1.8 Vdc.

Para la utilización de estos mecanismos en lenguaje C existen ciertas limitaciones que en realidad también se extienden en Java Script, pero que no se presentan en Python, pues



Phyton se puede utilizar de forma intuitiva como una terminal de uso directo del kernel de Linux, tal como se especificó en las descripciones anteriormente descritas.

En lenguaje C++ NO se puede hacer de esta forma y para ingresar a los subdirectorios del kernel de Linux se debe hacerlo mediante archivos, que manejen punteros que apunten a los mencionados subdirectorios, y lean desde o escriban hacia los contenidos de dichos subdirectorios, es así que los programas se basan en instrucciones de lectura y escritura de archivos randómicos en lenguaje C++. Tal como en el siguiente código de programa se puede observar.

```
FILE *PunARCHIVO = NULL;

if ((PunARCHIVO = fopen("/sys/class/gpio/export", "ab")) == NULL) {

printf("No se puede exportar ese pin gpio\n");

return 1;

}

strcpy(Valor, "38");

fwrite(&Valor, sizeof(char), 2, PunARCHIVO);

fclose(PunARCHIVO);
```

En donde se define el puntero al archivo y las instrucciones necesarias para escribir un dato en el archivo. Nótese que el puntero apunta a /sys/class/gpio/export, es decir al subdirectorio del kernel, y el dato que se escribe es "38", que equivale a escribir la instrucción directamente en el kernel: echo 38 > /sys/class/gpio/export. Es la forma que se ejecutan las instrucciones en lenguaje C++, verificando la poca eficiencia de escritura de los programas, pero de ninguna forma la potencia de programación del lenguaje.

Las funciones que se han creado para el manejo de la plataforma de producción y detección de la fuente de luz (odorante) se describen y explican a continuación:

- Para la producción de la fuente luminosa se han creado las siguientes funciones:
  - **int readADC(void):**

Es una función que toma los valores digitales que se producen a la salida del conversor analógico – digital que tiene cada entrada análoga y presenta una palabra digital de 12 bits transformada en decimal que va de entre 0 a 4095. Esta

se obtiene de la entrada del potenciómetro analógico conectado a ain0, y se convierte a su equivalente duty\_percent dentro de la placa Beaglebone mediante software poniendo en acción la siguiente fórmula:

$$m = \frac{100 - 0 (\%)}{4095 - 0}$$

$$4095 - Pot = m(100 - DP)$$

Es decir describe una relación lineal entre la variación de cada valor digital de entrada la relación de trabajo del sistema de modulación PWM.

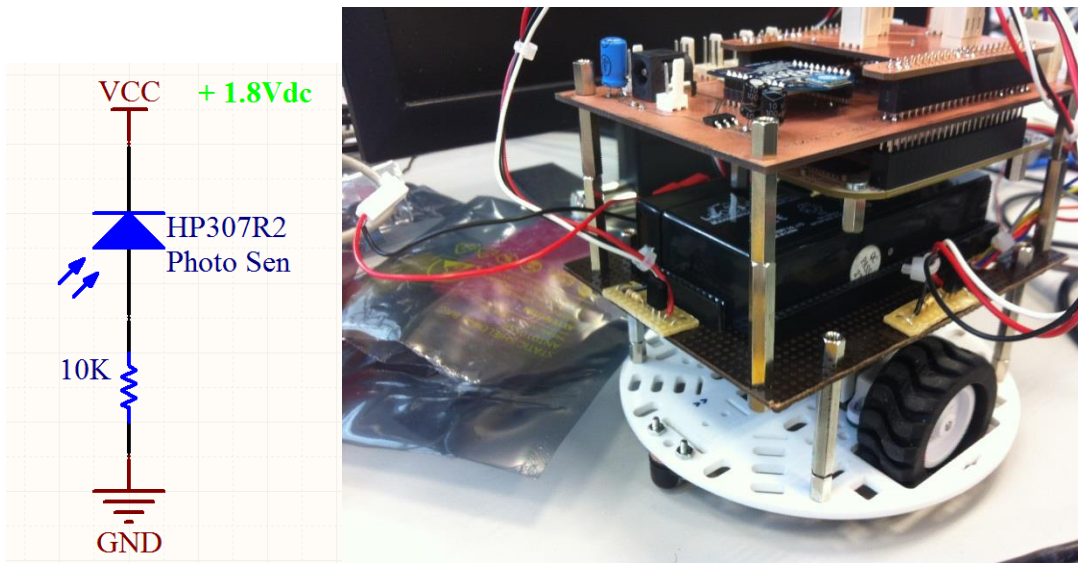
○ **int PWMPort(int GPIOPin):**

Esta función realiza algunos procedimientos:

1. El primero de ellos llevar el pin al modo 6, es decir al modo de trabajo PWM.
2. En segundo lugar, reservar ese pin para el trabajo PWM: request.
3. Definir el modo de trabajo por período o frecuencia.
4. Definir el porcentaje de la relación de trabajo del PWM.
5. Definir la frecuencia de la señal de salida.
6. Correr la aplicación de salida.

Esta aplicación ingresa la salida PWM en la base del MOSFET que se abre y cierra, variando así la potencia media entregada en la carga, que para este caso es el conjunto de 8 leds infrarrojos. Revisar Figura 4.30.

- Para la detección de la luz infrarroja se han utilizado 4 sensores en cada extremo del robot, que leen la variación de voltaje sobre una resistencia en serie conectada a un fotodiodo, que conduce total o parcialmente según la intensidad infrarroja que le llega a su superficie, haciendo que de esta forma varíe el voltaje de la resistencia con la misma variación del fotodiodo. La Figura 5.3 muestra el circuito esquemático y los sensores contruidos y montados en el robot.



**Figura 5.3 Diagrama de conexiones y conexión física de los detectores de luz infrarroja**

Para la lectura simultánea de los sensores de luz infrarroja dentro del robot, solamente se usó una función similar en contexto a la función “`int readADC(void)`” explicada anteriormente, pero en este caso se leen los puertos analógicos ain4, ain5, ain6 y ain7, designadas dentro del programa como sensor norte, sur, este y oeste.

Como se explicó anteriormente las señales analógicas de entrada de la placa Beaglebone son obtenidas a través de una palabra digital de 12 bits que se presenta directamente al invocar en el kernel de Ubuntu la sentencia “`cat ainX`” en el subdirectorio `/sys/devices/platform/tsc/`, donde X es el valor de la entrada analógica de 1 a 7. Con esta consideración es necesario definir las variaciones de voltaje que pueden ser detectadas por este sensor, de acuerdo a la siguiente fórmula:

$$resolución = \frac{V_{in}}{2^{N \text{ bits}}}$$

Es decir:

$$resolución = \frac{1.8 \text{ Vdc}}{2^{12}}$$

$$resolución = \frac{1.8V}{4096} \cong 439\mu V$$

Que aparentemente reflejan el nivel de detección que puede tener el sensor, pero no es así, ya que así solamente se está calculando la resolución del conversor A-D que tiene

internamente la placa Beaglebone, el hecho de que ella tiene esa capacidad de discriminación no significa que el sensor tenga esa capacidad de entrega de esos intervalos de voltaje tan pequeños, antes bien es al contrario, el sensor entregará la variación de voltaje correspondiente a la variación de distancia a la que se encuentra la fuente de luz infrarroja.

Lamentablemente se comprobó de forma experimental que los sensores que se utilizaron para el efecto, producían variaciones de tensión “detectables” por la entrada análoga de la placa Beaglebone cuando la fuente de luz se encontraba a máximo 10 cm de distancia del sensor, lo que hacía que el robot deba estar muy cercano a la fuente de luz para que lo pueda detectar, lo que constituye un gran problema para su eficiencia, pero de otro lado tenía una gran certeza de haber encontrado la fuente de luz y enviar su posición casi exacta al sistema de supervisión y sus compañeros.

Para la aplicación que se está realizando esta distancia es muy corta y presentaba muchos problemas el que el robot la encontrara. Afortunadamente en el HCTLab, se realizan experimentos con tecnologías electrónicas muy diversas y es así que poseen emisores de infrarrojos de alta potencia que nos dan la posibilidad de aumentar la distancia de detección del robot y aprovechar la enorme capacidad de la placa Beaglebone al leer tensiones de entrada tan pequeñas, es decir, intervalos de distancia más cercanos o lejanos, como se desee apreciar.

Es así que la configuración del emisor de luz infrarroja que se aprecia en la figura 4.30 se modificó de dos formas, en principio solamente se reemplazaron los sensores de luz infrarroja GaAIAs OP293 de OPTEK, por el SFH415 de OSRAM que presenta la posibilidad de irradiar luz infrarroja con una corriente circulante de 1A, a intervalos de frecuencia de 10nS, los elementos se aprecian en la Figura 5.4.



**Figura 5.4 Led emisores de infrarrojos utilizados.**

Pero manteniendo el circuito original, es decir las resistencias limitadoras y el MOSFET manejador de la señal PWM, ésta configuración mejoró en casi 10 veces la distancia de detección de tal forma que se aprecian señales luminosas desde 81 cm, en ambientes oscuros. La forma final de conexión se la presentará más adelante en la integración, pruebas y resultados, ya que consiste en el intento de detectar mayores distancias que esas, llevándole al límite al sensor cortocircuitando las resistencias y haciendo pasar 4 Acd por el MOSFET.



## **CAPITULO VI**

### **IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIONES Y LA COLABORACIÓN ENTRE AGENTES**





## 6 Implementación del sistema de comunicaciones y la colaboración entre agentes.

Beaglebone tiene la posibilidad de comunicarse con casi todos los sistemas más conocidos y utilizados actualmente, a saber, comunicación Ethernet, USB, serial, I<sup>2</sup>C, CAN, etc. Es decir presenta una versatilidad envidiable con respecto a este soporte.

El determinar la opción que más conviene al proyecto ha desembocado en dos aspectos fundamentales: primero en la sencillez y facilidad de enmascaramiento de los datos, es decir que no se tengan que utilizar muchas capas para servicio de las comunicaciones, mandarlos en banda base de ser posible y, la forma inalámbrica de conectarse, dada por la necesidad que el robot sea autónomo y se comunique eficientemente sobre un área reducida.

Con estas consideraciones se eligió XBee, porque en la transferencia de datos, desde la parte del programa, no se necesita definir ningún tipo de enmascaramiento de la información, menos aún cabeceras, ni colas que determinen su eficiente transporte y seguridad, es más no se requiere seguridad en los datos, más bien solamente se usa un subdirectorío del kernel que maneja el buffer de la UART y los datos se transfieren el momento que están disponibles en él, la sincronización de envío de los mismos se da únicamente por la iteración de programa que se está ejecutando, pudiendo aumentarla introduciendo períodos de espera (retardos) con medios estándar o personalizados, permitidos en la plataforma de lenguaje C++. Se habla de retardos porque no son requeridas comunicaciones de alta velocidad, que si bien es cierto las podría desarrollar la placa (recordar que trabaja a 750 MHz), pero no son necesarias, más aun se necesitan que sean algo lentas para darle al sistema de supervisión algo de coherencia y intuitividad requerida por el operador humano.

Lo dicho, para el sistema de comunicaciones se necesita un entorno sencillo, en donde los datos fluyan fácilmente por el canal y sean así mismo fácilmente reconocidos, tanto por el sistema software como por los demás agentes. La causa de este requerimiento es que no hace falta “cargar” el programa de usuario de los agentes con líneas de código para controlar el sistema, basta con transferir la localización continua de los agentes y si ha sido capaz de localizar la fuente de olor/luz.

Es por esta razón de que, a pesar de tener los módems XBee la posibilidad de usar varias líneas de operación (20), se usan apenas 4 de ellas, y dos para polarización, dando a lugar un canal de comunicaciones básico con un transmisor y un receptor que no tiene pérdidas

introducidas por cables, porque prácticamente se coloca en la fuente – destino de la información que es la placa Beaglebone.

Entonces como se mencionó en las secciones 4.11 y 4.17 el hardware contempla solamente las líneas de transmisión y recepción del modem XBee conectados directamente al header P9.24 y P.26 denominados UART1\_TXD y UART1\_RXD respectivamente. Hay que tener mucho cuidado con la interpretación que se le da a las señales que entran al módulo XBee y se conectan al UART de la placa Beaglebone, pues las señales de transmisión y reflexión se pueden confundir con las correspondientes a la antena de radiación de radiofrecuencia; es por esta razón que en muchos manuales de XBee se las representa como señal in y señal out, para no traslaparse con tx y rx, que se refieren a las de la antena.

En forma práctica la señal del pin P9.24 se conecta al pin 1.3 de XBee y las del pin P9.26 al pin 1.2 de XBee.

Para realizar la configuración de los módems, se tiene que identificar el tipo de modem disponible, pues, independientemente de sus características técnicas, existen tres tipos de módulos que son: Módems XBee, XBee PRO S1 (serie 1) y XBee PRO S2. Esta identificación es importante porque depende de ella los parámetros existentes en los módulos de configuración y los tipos de enlaces que pueden soportar cada uno de ellos.

## **6.1 Tipos de dispositivos configurables en una red con XBee.**

Una red Zigbee la forman básicamente 3 tipos de elementos. [11]. Un único dispositivo Coordinador, dispositivos Routers y dispositivos finales (end points).

### **6.1.1 El Coordinador.**

Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points. Una vez formada la red, el Coordinador hace las funciones de Router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

### **6.1.2 Los Routers.**

Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información. Lógicamente un router debe unirse a una red

Zigbee antes de poder actuar como Router retransmitiendo paquetes de otros routers o de End points.

### **6.1.3 End Device.**

Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Router, es decir, no puede enviar información directamente a otro end device. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

### **6.1.4 Aplicaciones.**

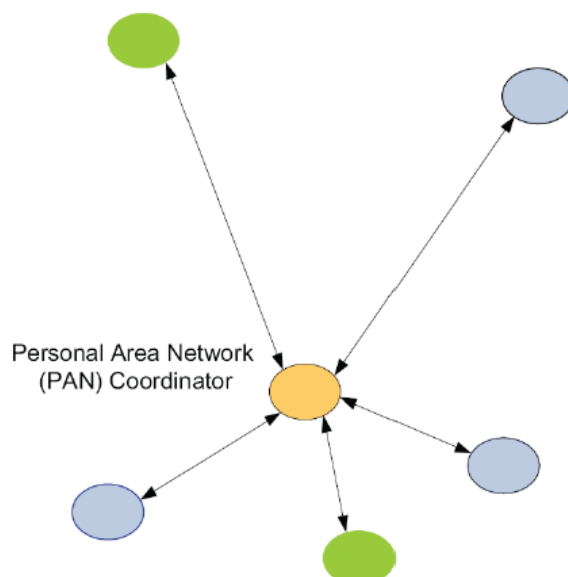
Cada módulo XBee (o Zigbee es la misma terminología cuando se habla de los módems), al igual que ocurre con las direcciones MAC de los dispositivos Ethernet, tiene una dirección única. En el caso de los módulos XBee cada uno de ellos tiene una dirección única de 64 bits que viene grabada de fábrica. Por otro lado, la red Zigbee, utiliza para sus algoritmos de ruteo direcciones de 16 bits. Cada vez que un dispositivo se asocia a una red Zigbee, el Coordinador al cual se asocia le asigna una dirección única en toda la red de 16bits.

Por eso el número máximo teórico de elementos que puede haber en una red Zigbee es de  $2^{16} = 65536$ , que es el nº máximo de direcciones de red que se pueden asignar.

Estos módulos XBee, pueden ser ajustados para usarse en redes de configuración punto a punto, punto a multipunto o peer-to-peer. Un ejemplo se muestra en la figura 6.1., donde se muestra una conexión multipunto, con un coordinador, conectado a varios nodos. DIGI® tiene 2 series de módulos disponibles. Los módulos XBee Serie 2 permite hacer redes tipo mesh, y los módulos XBee de la serie 1 no lo pueden hacer, pero en nuestro caso se utilizará un maestro que es el sistema de supervisión y muchos esclavos que son los entes de las celdas, porque de lo contrario, todos los robots deberían tener la capacidad computacional suficiente para resolver direcciones de red, identificarlas y darles a cada una de ellas el servicio deseado, que en muchos casos puede ser redundante con los demás nodos y entre ellos, aumentando innecesariamente el tráfico de la red y duplicando de la misma forma la información. Además de que el alcance máximo de 30 metros es más que suficiente para el entorno en que se pretende que el proyecto prospere.

También existen los llamados módulos XBee PRO de la Serie 1 que se diferencian en la capacidad de alcance, permitiendo en algunos casos doblar la distancia de transmisión, ya que poseen una mayor potencia en la señal. Con los módulos XBee PRO de la Serie 2, es posible crear redes más complejas, como las llamadas MESH. Estas permiten acceder a un

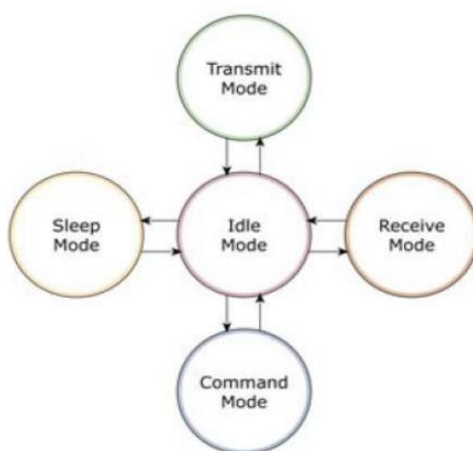
punto remoto, utilizando módulos intermedios para llegar como routers. Además los módulos automáticamente generaran la red entre ellos, sin intervención humana alguna, permitiendo la reparación de la red en caso de que algún nodo falle. Al mismo tiempo la red por sí sola resuelve la mejor ruta para un determinado paquete.



**Figura 6.1 Red Multipunto con un solo Módulo Coordinador. Tomada de la Guía de usuario XBee del enlace [www.olimex.cl](http://www.olimex.cl)**

### 6.1.5 Modos de Operación.

Los módulos XBee pueden operar en los siguientes cinco modos:



**Figura 6.2 Red Multipunto con un solo Módulo Coordinador. Tomada de la Guía de usuario XBee del enlace [www.olimex.cl](http://www.olimex.cl)**

#### **6.1.5.1 Modo RECIBIR/TRANSMITIR.**

Se encuentra en estos modos cuando al módulo le llega algún paquete RF a través de la antena (modo Receive) o cuando se manda información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit).

La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita.

Además es posible enviar información por dos modos. Unicast y Broadcast. Por el primero, la comunicación es desde un punto a otro, y es el único modo que permite respuesta de quien recibe el paquete RF, es decir, quien recibe debe enviar un ACK (paquete llamado así, y que indica que recibió el paquete, el usuario no puede verlo, es interno de los módulos) a la dirección de origen. Quien envió el paquete, espera recibir un ACK, en caso de que no le llegue, reenviará el paquete hasta 3 veces o hasta que reciba el ACK. En el modo Broadcast la comunicación es entre un nodo y a todos los nodos de la red. En este modo, no hay confirmación por ACK.

#### **6.1.5.2 Modo de Bajo Consumo (Sleep Mode).**

El modo de sueño hace posible que el módulo RF entre en un modo de bajo consumo de energía cuando no se encuentra en uso.

Para poder entrar en modo de sueño, se debe cumplir una de las siguientes condiciones:

- Sleep\_RQ (pin 9) está en alto y el módulo está en pin sleep mode; (SM= 1,2 o 5).
- El módulo está en reposo (no hay transmisión ni recepción de datos) por la cantidad de tiempo definido por ST (Time before Sleep). {ST sólo está activado cuando SM=4,5}.

La configuración de los ciclos de sueño se realiza principalmente con el comando SM. Por defecto, los modos de sueños están deshabilitados (SM=0), permaneciendo el módulo en estado de reposo/recepción. En este estado el módulo está siempre preparado para responder a un comando, ya sea, por el puerto serial o la interfaz RF.

#### **6.1.5.3 Modo de Comando.**

Este modo permite ingresar comandos AT al módulo XBee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es necesario utilizar el Hyperterminal de Windows, el programa X-CTU o

algún mecanismo inteligente que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.

Para ingresar a este modo se debe esperar un tiempo dado por el comando GT (Guard Time, por defecto ATGT=0x3E84 que equivalen a 1000ms) luego ingresar +++ y luego esperar otro tiempo GT. Como respuesta el módulo entregará un OK. El módulo XBee viene por defecto con una velocidad de 9600bps. En caso de no poder ingresar al modo de comandos, es posible que sea debido a la diferencia de velocidades entre el módulo y la interfaz que se comunica vía serial.

#### **6.1.5.4 Modo Transparente**

En este modo todo lo que ingresa por el pin 3 (Data in), es guardado en el buffer de entrada y luego transmitido y todo lo que ingresa como paquete RF, es guardado en el buffer de salida y luego enviado por el pin 2 (Data out). El modo Transparente viene por defecto en los módulos XBee.

Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control. También se usa para reemplazar alguna conexión serial por cable, ya que es la configuración más sencilla posible y no requiere una mayor configuración.

En este modo, la información es recibida por el pin 3 del módulo XBee, y guardada en el buffer de entrada. Dependiendo de cómo se configure el comando RO, se puede transmitir la información apenas llegue un carácter (RO=0) o después de un tiempo dado sin recibir ningún carácter serial por el pin 3. En ese momento, se toma lo que se tenga en el buffer de entrada, se empaqueta, es decir, se integra a un paquete RF, y se transmite. Otra condición que puede cumplirse para la transmisión es cuando el buffer de entrada se llena, esto es, más de 100 bytes de información.

#### **6.1.5.5 Modo de operación API**

Este modo es más complejo, pero permite el uso de frames con cabeceras que aseguran la entrega de los datos, al estilo TCP. Extiende el nivel en el cual la aplicación del cliente, puede interactuar con las capacidades de red del módulo.

Cuando el módulo XBee se encuentra en este modo, toda la información que entra y sale, es empaquetada en frames, que definen operaciones y eventos dentro del módulo.

Así, un Frame de Transmisión de Información (información recibida por el pin 3 o DIN) incluye:

- Frame de información RF transmitida.

- Frame de comandos (equivalente a comandos AT).

Mientras que un Frame de Recepción de Información incluye:

- Frame de información RF recibida.
- Comando de respuesta.
- Notificaciones de eventos como Reset, Associate, Disassociate, etc.

Esta API, provee alternativas para la configuración del módulo y ruteo de la información en la capa de aplicación del cliente. Un cliente puede enviar información al módulo XBee. Estos datos serán contenidos en un frame cuya cabecera tendrá información útil referente el módulo.

Esta información además se podrá configurar, esto es, en vez de estar usando el modo de comandos para modificar las direcciones, la API lo realiza automáticamente. El módulo así enviará paquetes de datos contenidos en frames a otros módulos de destino, con información a sus respectivas aplicaciones, conteniendo paquetes de estado, así como el origen, RSSI (potencia de la señal de recepción) e información de la carga útil de los paquetes recibidos.

- Entre las opciones que permite la API, se tienen:
- Transmitir información a múltiples destinatarios, sin entrar al modo de Comandos.
- Recibir estado de éxito/falla de cada paquete RF transmitido.
- Identificar la dirección de origen de cada paquete recibido.

#### **6.1.5.6 Idle.**

Cuando el módulo no se está en ninguno de los otros modos, se encuentra en éste. Es decir, si no está ni transmitiendo ni recibiendo, ni ahorrando energía ni en el modo de comandos, entonces se dice que se encuentra en un estado al que se le llama IDLE.

## **6.2 Configuración de los módulos XBee.**

Para la configuración de los módulos XBee de cualquier serie y modo de operación, se lo realiza por uno de los tantos medios fabricados por el constructor DIGI®, para el efecto, que van desde un módulo sencillo llamado USB Explorer hasta módulos con sensores y dispositivos listos para simulación y conexión de datos a los módems.

El Laboratorio HCTLab, cuenta con dos dispositivos para el efecto, el XBee USB Explorer XBIB-U y el XBee serial Explorer XBIB-R que se observan en la Figura 6.3.

Estos módulos configuradores poseen características especiales que permiten a más de comunicarse de la forma estándar con el ordenador, es decir USB o RS232, implementar señales de prueba del dispositivo, que puede configurar y utilizar puertos de I/O de señales digitales y señales analógicas, de una manera cuasi transparente al módem, es decir configurar adecuadamente las señales que serán enviadas por la antena que pueden tener diversas características.

En el caso del robot las señales a enviarse son datos digitales sin enmascaramiento, únicamente su representación ASCII y esa es la que se lee en el sistema de supervisión. En forma general debe enviar, como requisito, las coordenadas del agente del cuadrante  $n$ , es decir dos datos: el valor de  $x$  e  $y$ ; además como se ha planteado la existencia de una superficie superior dividida en semiplanos más pequeños que son los que supervisan los robots, se debe identificar al agente, es decir indirectamente el semiplano en el que se encuentra, en respuesta el sistema de supervisión deberá contextualizar la localización general e interactuar con cada agente para llevarle a la conclusión del objetivo común.



**Figura 6.3 Módulos de configuración seriales vía RS232 y USB para módems XBee.**

Después de conectar en los receptáculos correspondientes al modem XBee (de cualquier modelo), se debe reconocerlos a través de software. En primer lugar, tendrá que disponerse e instalarse los drivers de manejo del dispositivo Explorer, que se pueden obtener el momento de la adquisición de los módulos o descargarse de la página oficial del fabricante: <http://www.digi.com/support/productdetail?pid=3140&osvid=0&type=drivers>.

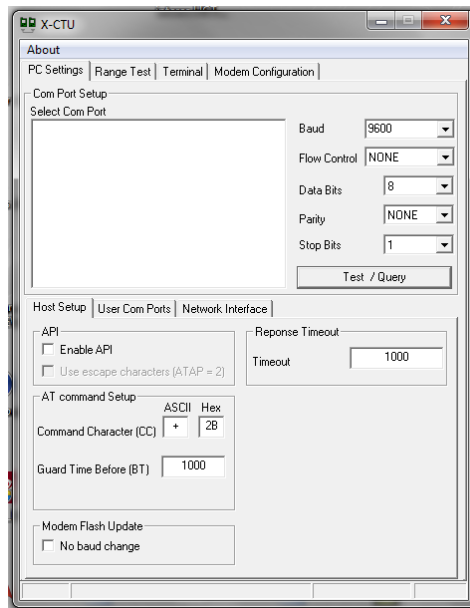
La configuración puede hacerse utilizando los comandos en la línea de texto del hiperterminal (para Windows 7 deberá descargarse la aplicación, ya que no está más disponible), caso para el que, deberá conocerse todos ellos y las respuestas que emiten al introducirlos.

El segundo método es hacerlo utilizando el software facilitado por DIGI® que se llama X-CTU, el cual se lo puede descargar del enlace:



<http://www.digi.com/support/kbase/kbaseresultdetl?id=2125>

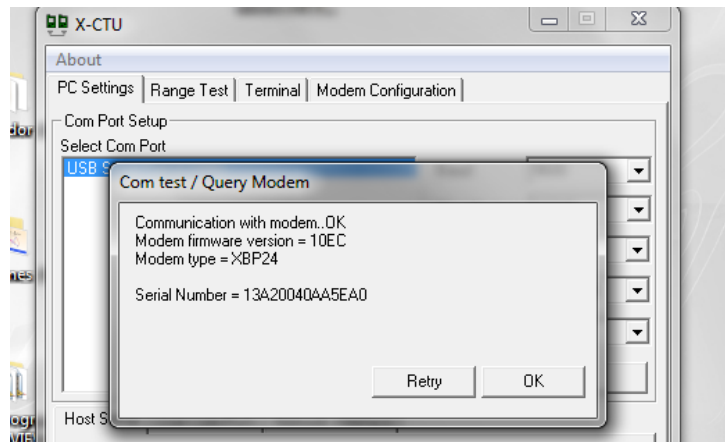
Este es un programa ejecutable que sube el instalador para Windows 7; Se deberá seguir todas las instrucciones de instalación y después de ello, ejecutarlo con doble clic en el ícono que se crea en el escritorio del ordenador. Al hacerlo se establece la interfase que se observa en la Figura 6.4



**Figura 6.4 Pantalla de inicio de software X-CTU para configuración de XBee.**

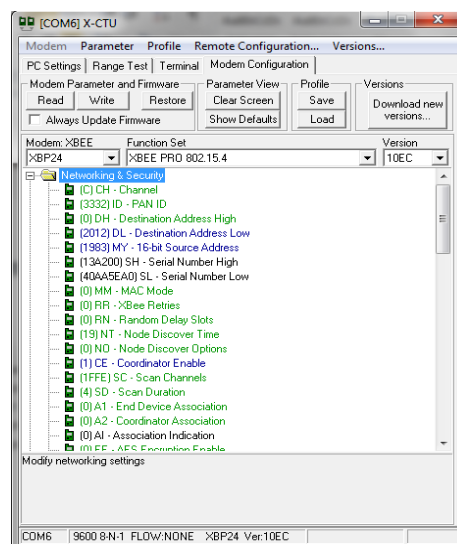
En este configurador se siguen los siguientes pasos:

1. Comprobar la conectividad del puerto.- Como se está haciendo una conexión USB o serial, se comprobará la conexión, que se establecerá automáticamente siempre y cuando los drivers hayan sido instalados, esto puede realizarse en: panel de control >> administrador de dispositivos >> Puertos COM y LPT. Se identificará el número del COM que se crea; en el caso del configurador serial, residirá en el COM1 a menos que algún otro dispositivo lo esté utilizando.
2. Prueba del puerto.- Presionando el botón Test / Query se establece la comunicación con el Explorer y se verifica porque devuelve el modelo y serie del modem conectado en él (se recomienda anotar ese código porque en algunas configuraciones se sugiere la conexión de la red con los códigos alto y bajo de los módems). Esto se muestra en la Figura 6.5



**Figura 6.5 Reconocimiento del modem conectado al Explorer mediante X-CTU.**

3. Luego de los cual se podrá configurar el modem y monitorearlo en las pestañas: Modem Configuration y Terminal.
4. En Modem Configuration, se establecerán las condiciones de operación del modem conectado al receptáculo del explorador. En primer lugar hay que reconocer cuál es su tipo; presionando el botón READ; la respuesta del software es en primer lugar el modelo del modem y las funciones para las que está configurado, de fábrica o después de una o varias configuraciones, éstas son las configuraciones que están establecidas en el modem y pueden ser cambiadas de acuerdo a las necesidades de la conexión o aplicación; posteriormente grabados en dicho elemento, esto se podrá hacer utilizando las opciones READ, WRITE Y RESTORE. Presionando el botón SHOW DEFAULTS se verá la configuración de fabricante para el módulo de ese modelo. Esto se verifica en la Figura 6.6.



**Figura 6.6 Opción de configuración de módem XBee**

Después de realizar las configuraciones necesarias, el modem está listo para recibir y transmitir la información sin necesidad de estar conectado al Explorer, con cualquier hardware relacionado, podría ser un microcontrolador o cualquier sistema embebido como un Arduino, Raspberry y en nuestro caso Beaglebone o Black Beaglebone.

El requisito para esto es solamente polarizar al módulo, es decir funciona en el modo transmitir / recibir que se ha mencionado en la sección 6.1.

Para la configuración de los dos módulos XBee, uno se colocará en el robot y otro irá en un receptáculo de un Explorer USB o RS232, se debe tener la idea clara de los siguientes aspectos:

1. El nodo del sistema de supervisión debe configurarse como Coordinador.
2. La red debe ser la misma, tener el mismo PAN ID
3. Las direcciones Alta y baja de la red, deberá corresponder a los códigos seriales de cada módulo XBee que se utilice.
4. La potencia de transmisión se elegirá por la distancia de cobertura. Se recomienda de ALTA y si el alcance sería extremo (cosa que no permitirá la “nariz” eventualmente) se configurará como MUY ALTA.

Para el desarrollo del prototipo, se ha considerado únicamente la comunicación de un nodo coordinador y un End device, es decir comunicación punto a punto; por dos aspectos importantes:

- Solo existe un robot y el sistema de supervisión centralizado en el computador, es decir es una conexión punto a punto simple.
- Para la inserción de más nodos y hacer conexión punto a multipunto se adicionará únicamente el PAN ID a todos los nodos y obviamente la dirección del coordinador.

La conexión punto a punto es la conexión ideal para reemplazar comunicación serial por un cable. Sólo se debe configurar la dirección. Para ello se utilizan los comandos MY y el DL. La idea, es que se define arbitrariamente una dirección para un módulo, usando el comando MY, el cual se va a comunicar con otro que tiene la dirección DL, también definida arbitrariamente. Con esto cada módulo define su dirección con MY, y escribe la dirección del módulo al cual se desea conectar usando DL. Que son valores personalizados de nombre de nuestra propia red.

En este modo, el módulo receptor del mensaje envía un paquete al módulo de origen llamado ACK (Acknowledge) que indica que el mensaje se recibió correctamente.

Para la conexión Punto a Multipunto,

Tomando en consideración todos estos aspectos, la configuración final de los módems que posee HCTLab que tienen número de serie 0013A20040AA5EA0 y 0013A20040AA5EB8 se ha hecho con los siguientes parámetros:

Para el nodo que irá conectado al receptáculo del XBee Explorer:

- Tipo de modem XBP24 se da por defecto en la lectura.
- Function set: XBEE 802.15.4 (es seleccionable, pero aparece por defecto).
- Coordinator enable: 1 COORDINATOR.
- DL 2013 (dirección baja del destino).
- MY 1983 (dirección de la fuente).
- POWER LEVEL: 3 HIGH.
- Sample Rate: 3F8 (1 seg)
- I/O Input Address: FFFF

Para el nodo conectado al robot:

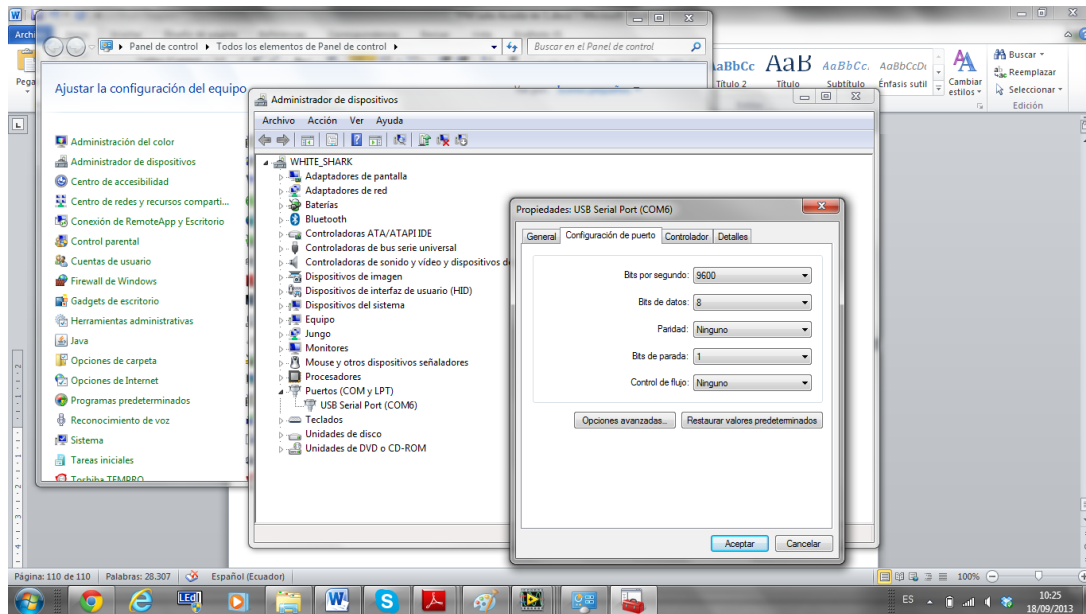
- Tipo de modem XBee: XBP24
- Function set: XBEE 802.15.4
- Coordinator enable: 0 End Device.
- Power Level: 3 HIGH
- Sample rate: 3F8 (1 seg)
- D3 configuration: 3DI
- D2 configuration: 2DI
- D1 configuration: 1DI
- D0 configuration: 0DI
- IT: 1
- Input address: FFFF

Realizadas las pruebas correspondientes, la información “valiosa” está contenida en el byte 11 de la trama de datos que llega, y se discrimina y presenta en el sistema de supervisión.

### **6.3 Implementación del hardware y software del sistema XBee.**

Ahora ya estando listos los módulos, en el lado del computador se deberá desarrollar el software de supervisión, visto que los datos ya están disponibles en el puerto serial, es decir la única consideración de hardware que hay que hacer es verificar la existencia del puerto serial virtual y su denominación.

Eso se verifica en inicio >> panel de control >> administrador de dispositivos >> dispositivo COM y LPT. Y se muestra en la Figura 6.7.



**Figura 6.7 Verificación del puerto serial.**

En el robot en cambio, solamente se conectará en el sócalo construido para ello ya que está ruteada la conexión entre el pin 3 y la salida de datos de Beaglebone y el pin 2 y la entrada de datos de Beaglebone; además de que se ha conectado también la polarización de 3.3Vdc.

Como ya está configurado el modem y está en modo transparente, solamente deberán enviarse los datos por el buffer, es decir enviarlos hacia el pin 3, esto corresponde a enviar datos por el header 24 de la placa Beaglebone.

Como en los casos anteriores, los pines del header Beaglebone se deben configurar para llevarles al modo correspondiente (excepto ainX que se debe solamente verificar); como se puede observar en las tablas 4.5, 4.6 y 4.7, los modos de operación del pin P9.24 y P9.26 son varios, es decir hay que llevarlos para que sean operativamente UART1\_TXD y UART1\_RXD respectivamente, esta operación como en los casos mencionados anteriormente se verifica primero el estado de dichos pines y luego se les cambia el modo si es necesario; en el programa principal del robot, se determina en cada función de operación primero esta configuración, no se ha utilizado la idea inicial de programación de configurar todos los pines en una sola función inicial, porque muchas veces se necesita particularizar esta operación, como se verá más adelante, en el capítulo siguiente. Los procedimientos de operación son para el pin P9.26:

- root@beaglebone:~# cat /sys/kernel/debug/omap\_mux/uart1\_rxd
- name: uart1\_rxd.gpio0\_14 (0x44e10980/0x980 = 0x0037), b NA, t NA
- mode: OMAP\_PIN\_INPUT\_PULLUP | OMAP\_MUX\_MODE7
- signals: uart1\_rxd | mmc1\_sdwp | d\_can1\_tx | i2c1\_sda | NA | pr1\_uart0\_rxd | NA | gpio0\_14

Que indica el estado del Puerto por defecto en el modo 7, es decir gpio0\_14, y hay que trasladarlo al modo 0, según lo que se referencia en la tabla 4.6.

Pare el pin P9.24 la opción es:

- root@beaglebone:~# cat /sys/kernel/debug/omap\_mux/uart1\_txd
- name: uart1\_txd.gpio0\_15 (0x44e10984/0x984 = 0x0037), b NA, t NA
- mode: OMAP\_PIN\_INPUT\_PULLUP | OMAP\_MUX\_MODE7
- signals: uart1\_txd | mmc2\_sdwp | d\_can1\_rx | i2c1\_scl | NA | pr1\_uart0\_txd\_mux1 | NA | gpio0\_15

De igual forma el estado del pin por defecto es el modo 7 es decir gpio0\_15, debe llevarse al modo 0.

Se pensaría que hay que llevarlos con el valor 0 a los dos pines pero no es así, investigando en el manual de referencia de ARM para el procesador CORTEX-A8 a pesar de que los bits menos significativos tiene que ser cero, para habilitar la funcionalidad de los UART Tx y Rx se debe configurar el MODO 0 y, el bit 5 se configura como 1 en Rx y como 0 en Tx. Eso se observa en la tabla 6.1. tomada del manual de referencia ARM CORTEX A8.

**Tabla 6.1 Valor de los bits para manejo de TX y RX en ARM CORTEX, (tomado del manual de operación del fabricante)**

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-7	Reserved	R	0h	
6	conf_<module>_<pin>_slewctrl	R/W	X	Select between faster or slower slew rate 0: Fast 1: Slow Reset value is pad-dependent.
5	conf_<module>_<pin>_rxactive	R/W	1h	Input enable value for the PAD 0: Receiver disabled 1: Receiver enabled
4	conf_<module>_<pin>_pullupsel	R/W	X	Pad pullup/pulldown type selection 0: Pulldown selected 1: Pullup selected Reset value is pad-dependent.
3	conf_<module>_<pin>_pullupden	R/W	X	Pad pullup/pulldown enable 0: Pullup/pulldown enabled 1: Pullup/pulldown disabled Reset value is pad-dependent.
2-0	conf_<module>_<pin>_muxmode	R/W	X	Pad functional signal mux select. Reset value is pad-dependent.

En este caso las instrucciones para realizar este procedimiento son:

- `echo 20 > /sys/kernel/debug/omap_mux/uart1_rxd`
- `echo 0 > /sys/kernel/debug/omap_mux/uart1_txd`

Es decir 0010 0000<sub>16</sub> para el RX (bit 5 en 1) y 0000 0000<sub>16</sub> para el TX.

El programa interno en el robot, no configura velocidad, ni protocolo serial, porque está por defecto configurado en:

- baud rate: 9600.
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: none

Es decir la configuración del puerto serial por defecto, **ESTO EN LA PLACA BEAGLEBONE UART1**; para el desarrollo del sistema de supervisión debe configurarse el sistema a la misma trama, es decir el COMX del PC y del software que se utiliza.

Entonces solamente es entregar datos al buffer de salida mediante UART1\_TXD, pin P9.24 y recibirlos por UART1\_RXD pin P9.26, la velocidad de ésta lectura y escritura está definida por cada vez que se emplean las instrucciones echo y cat en la función, personalizable por retardos de tiempos programados. Dentro de la función se han utilizado el siguiente formato de manejo del UART y transparentemente XBee (ya que está configurado en el modo transparente y cada vez que tiene datos los envía por su antena):

- Para escribir el dato en el buffer echo "dato" > tty1.
- Para leer el dato del buffer cat tty1.

## 6.4 Sistema de supervisión.

Un sistema de supervisión se hace necesario para que el mecanismo de colaboración se produzca entre los agentes, pues de no existirlo la alternativa sería que cada uno de ellos tenga la suficiente capacidad computacional para resolver su trabajo en red y supervisar la red y enrutar los paquetes, la verdad es que Beaglebone tiene la capacidad de soporte para hacerlo, incluso en otro tipo de red, pero no se ha considerado así en este trabajo por las limitaciones del tiempo y complejidad del sistema, y a pesar que no se contemplaba en los objetivos del proyecto el diseño del sistema de supervisión, se ha desarrollado un prototipo para observar y dejar la plataforma lista para trabajos futuros, y principalmente es necesario para producir el trabajo colaborativo entre agentes.

El caso actual es que se tiene una red con un maestro que es un ordenador central, en el cual se ejecuta un programa desarrollado en un software de alto nivel, y muchos esclavos que dan y reciben la información. Cada uno de estos esclavos es un robot de características idénticas que realiza su trabajo de detección y localización de la fuente de “olor” en un área específica asignada a él, en este caso es una superficie cuadrada de 2x2 metros (así está hecho el programa principal del robot); cada robot manda permanentemente una información a este sistema de supervisión, esta consiste en sus coordenadas dentro de ese cuadrado, técnicamente los valores que envía el robot son: distancia de X, distancia de Y, y número de celda; entonces el sistema lo identifica y resuelve su localización exacta; cuando el robot envía continuamente una localización diferente de su referencia original, quiere decir que se detuvo y encontró la fuente de olor, en ese caso se debe comunicar a los demás agentes la misma y dirigirlos haciendo que se muevan cuanto antes hacia allá.

Actualmente la plataforma está desarrollada hasta el hecho de que el robot manda los datos y el sistema los recibe, se visualiza la recepción pero nada más, a lo mejor si alguien se interesa se puede hacer el sistema completo. En lo particular, como autor de este trabajo continuaré en esa investigación y muchas otras probables de la plataforma y todo el sistema que he desarrollado.

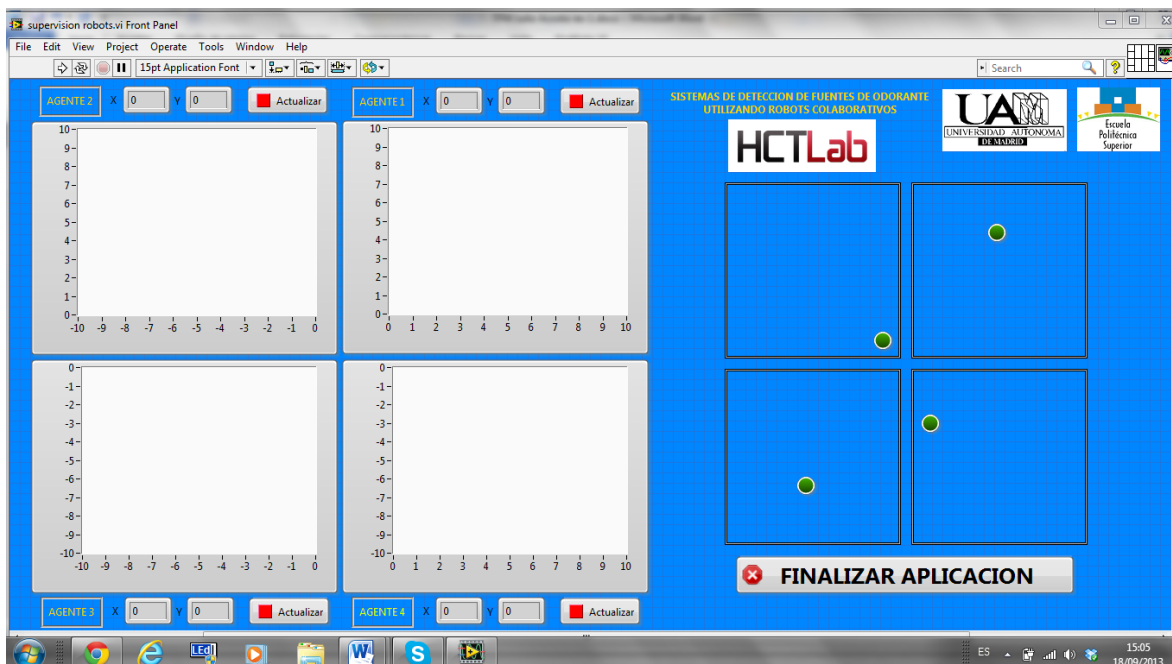
El software elegido para desarrollar el sistema es LabVIEW, porque se considera bastante versátil para este tipo de plataformas (gráficas y de supervisión), en él se puede desarrollar fácil e intuitivamente una plataforma gráfica que es lo que se necesita más que otra cosa (sin dejar de lado la inteligencia que debe poseer), y es así que posee extensas herramientas matemáticas, de procesamiento de señales, de comunicaciones, de control, etc. Que permiten que sea perfecto para el trabajo con la plataforma que se puede desarrollar.

Se ha construido una interfase gráfica que presenta un gráfico de coordenadas con 4 cuadrantes en los que se “supone” están cuatro robots, y que presentan su estado y su posición con cada iteración de programa, las posiciones se presentan continuamente con un gráfico de función  $y = f(x)$ , además de que al mismo tiempo se visualiza numéricamente esta posición; también se presentan las posiciones en un plano mediante luces indicadoras, de color verde y moviéndose para el caso de estar en estado de búsqueda y con una luz roja inmóvil cuando ha logrado dicha búsqueda.

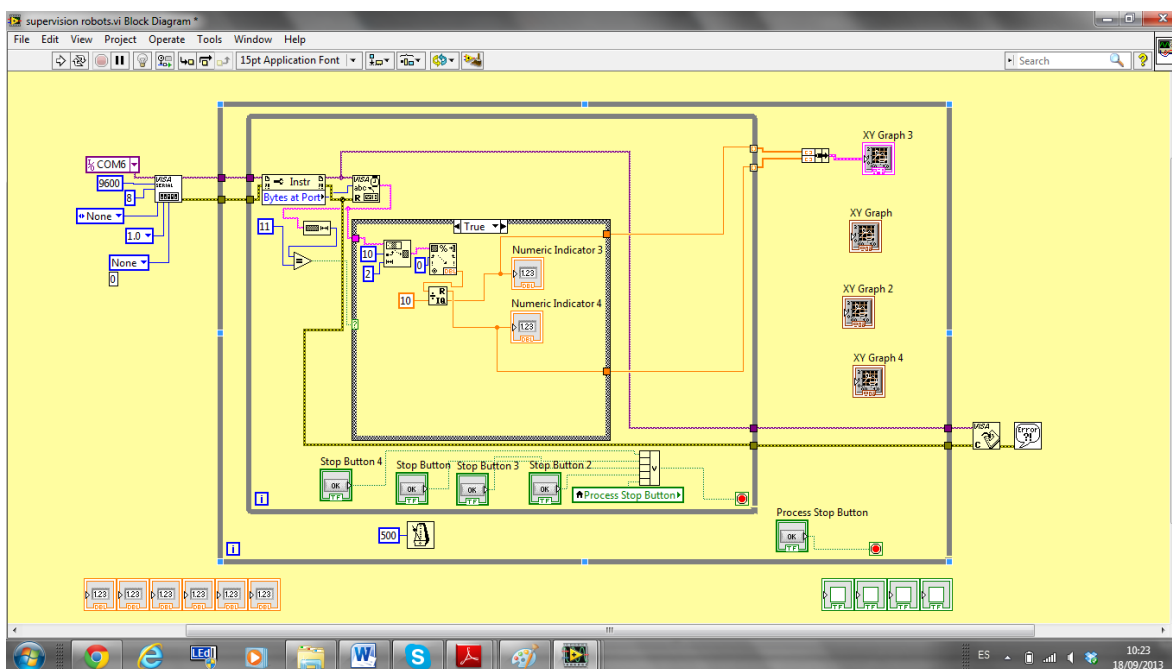
La interfase que se ha desarrollado se presenta en la Figura 6.8y en la Figura 6.9 se presenta el código gráfico que se ha utilizado para la programación de la interfase, se nota la capacidad de programación instalada y la disponibilidad de herramientas necesarias



para avanzar a partir de ella con especificaciones más técnicas y científicas de la operación de la interfase. El límite solamente la creatividad de quien lo desarrolle.



**Figura 6.8** Interfase grafica de la aplicación del sistema de supervisión de posición de los robots buscadores de odorante.



**Figura 6.9** Código de programación gráfico de la interfase de sistema de supervisión de posición de los robots buscadores de odorante.

A continuación se describe el código de programación utilizado:

- La primera instancia es la configuración del puerto, habiendo verificado el número del puerto creado el momento que se ha conectado el cable USB o el conversor USB – RS232, se configura con la herramienta VISA configure serial port, y se lo hace con los valores de defecto del puerto COMX: 9600, 8, n, 1, n, es decir 9600 baudios de velocidad, 8 bits de datos, y un bit de parada, nada más. Esta configuración muchas veces parece innecesaria porque ya se configura el puerto en el panel de control, pero puede ser que en otras aplicaciones se la haya modificado, entonces se recomienda ejecutarla para asegurar su operación.
- Luego la instrucción bytes at port, determina exactamente la cantidad de datos que ha recibido el buffer y no elimina información de manera indiscriminada por la longitud de la palabra que se desee leer.
- Luego la instrucción de lectura del puerto que se hace del resource name que no es otra cosa que el identificador del recurso hardware que se está utilizando. La salida de este módulo es una cadena string que contiene toda la trama de datos.
- Experimentalmente se determinó que la información adicional de la trama que se envía por los módulos, es decir la información “útil” para el sistema de colaboración está en el byte 11, es por eso que de toda la palabra string se toma dos caracteres a partir del carácter 11 y se lo convierte a un número decimal, que es el que contiene la información de coordenadas del robot. Desde este se envía una palabra compuesta de dos datos que contienen el valor de las coordenadas x e y en términos de un solo número, por ejemplo la coordenada P = (3,4) se representa por el número 34, por eso ya cuando se ha convertido a número la palabra string se le divide para 10 obteniendo un cociente y un residuo, el cociente es el valor de la coordenada X y el residuo es el valor de la coordenada Y. No se ha resuelto el valor de la dirección del agente porque se tiene uno solo (el programa está considerado como conexión punto a punto).
- Los valores obtenidos de x e y, se almacenan en dos vectores de 10 posiciones los cuales se van reciclando y cuando se presiona el botón de actualizar, se presentan en el gráfico del cuadrante correspondiente. En el caso particular está implementado para un solo agente del que se está verificando los valores enviados; si se incrementan los agentes, primeramente se deberá leer en la trama cual agente es el que se considera para poder programar el gráfico correspondiente.
- Por otra parte se está disponiendo también de 8 luces indicadoras de color verde, cada una de ellas representa a un robot colocado en un cuadrante, se les ha dado la ubicación pero su movimiento no está implementado porque se deberá hacer una personalización del objeto, lo que se llama en LabVIEW la variación de los nodos de propiedad, pero se puede hacer es por eso que se deja ya la idea de la funcionalidad

considerada; estos movimientos deberán coincidir en cuadrante y valor son los enviados por los agentes.

- Todo lo demás es redundante con los valores de cada agente identificado en el puerto de forma serial y continua. Para este programa se ha dado un retardo de 1Seg, que no tienen nada que ver con la velocidad que manda los datos el robot, sino con la capacidad del operador humano del sistema de supervisión de ver la información y tomar la acción correspondiente.
- El envío de datos a los agentes desde el sistema de supervisión no está resuelto, pero como es el mismo programa se deberá utilizar una estructura de secuencia, para en el mismo tiempo, no se necesita una velocidad muy elevada, primero leer el dato del buffer, y después ya liberados estos datos, escribir en él.



## **CAPITULO VII**

### **INTEGRACIÓN, PRUEBAS Y RESULTADOS**



## 7 Integración, pruebas y resultados.

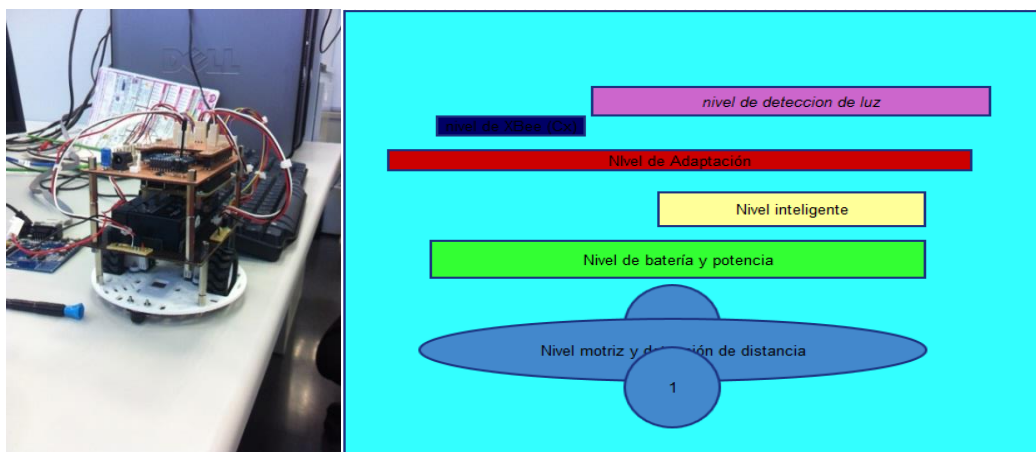
### 7.1 Integración del hardware del sistema y desarrollo del software.

Como se ha visto en las secciones anteriores, el hardware se ha construido con una visión común que es la de montarlo en forma central en la placa adaptadora y a partir de allí tender cables a todos los conectores de diferentes tipos que contienen tanto elementos electrónicos como mecánicos del robot.

Para ello se ha dividido la plataforma en cinco secciones o niveles de existencia de los módulos que son:

1. Sección motriz y de lectura de distancia y dirección.- en donde se encuentran ruedas y encoders.
2. Sección de soporte de potencia y alimentación.- en donde se encuentra la batería, sus conexiones y soporte.
3. Sección de dispositivo inteligente.- en donde se encuentra la placa Beaglebone.
4. Sección de adaptación y acople. En donde se encuentra la placa adaptadora y sus conexiones.
5. Sección de comunicaciones.- en donde se encuentra el módulo XBee.
6. Sección de detección de luz. En donde se encuentran los receptáculos de cada sensor de luz infrarroja, porque los sensores se tuvieron que bajar a nivel de la batería porque era la altura exacta que coincidía con la torre de led's infrarrojos construida.

Esto se observa en la Figura 7.1

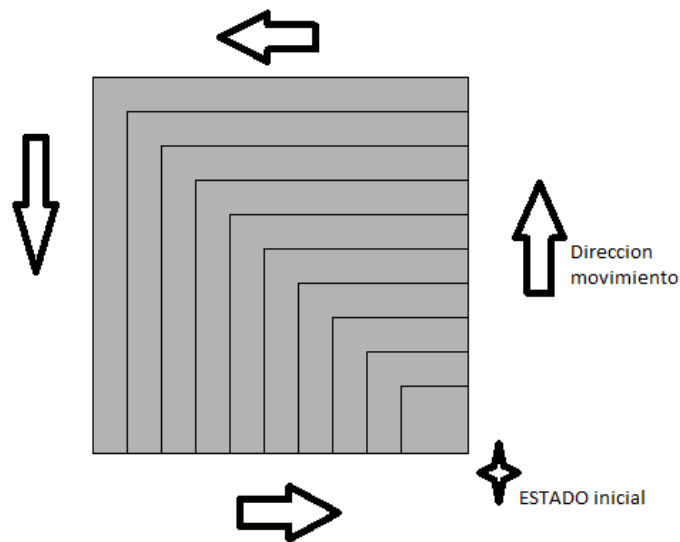


**Figura 7.1 Niveles de integración del hardware del Robot.**

## 7.2 Funciones software de los distintos niveles.

Dentro del programa principal para manejo del agente, si bien es cierto que se programaron y probaron cada uno de los módulos de hardware antes mencionados de formas “separadas” todos ellos se integran en un programa único y autónomo que realiza la llamada a cada función dependiendo de la secuencia de movimiento, detección de olor (luz) y comunicaciones y las decisiones que debe tomar en cada caso.

Para esto, se debe recordar la estrategia que emplea el robot para detección de la fuente de luz (olor). La estrategia es avanzar en línea recta desde el punto de coordenadas (0,0) y hacerlo a un valor de distancia máxima, luego de llegar a ella, girar 90° a la izquierda y recorrer nuevamente esa distancia, girar otros 90° y recorrer otra vez esa distancia, finalmente girar los últimos 90° y recorrer la distancia en su movimiento final. De esta forma ha descrito un cuadrado perfecto de valor de lado igual a distancia (en el robot construido corresponde a 2m). Luego reduce en un factor constante esa distancia y vuelve a hacer el recorrido; así sucesivamente hasta encontrar el punto inicial que coincida con el final. La Figura 7.2 muestra la trayectoria que persigue la estrategia:

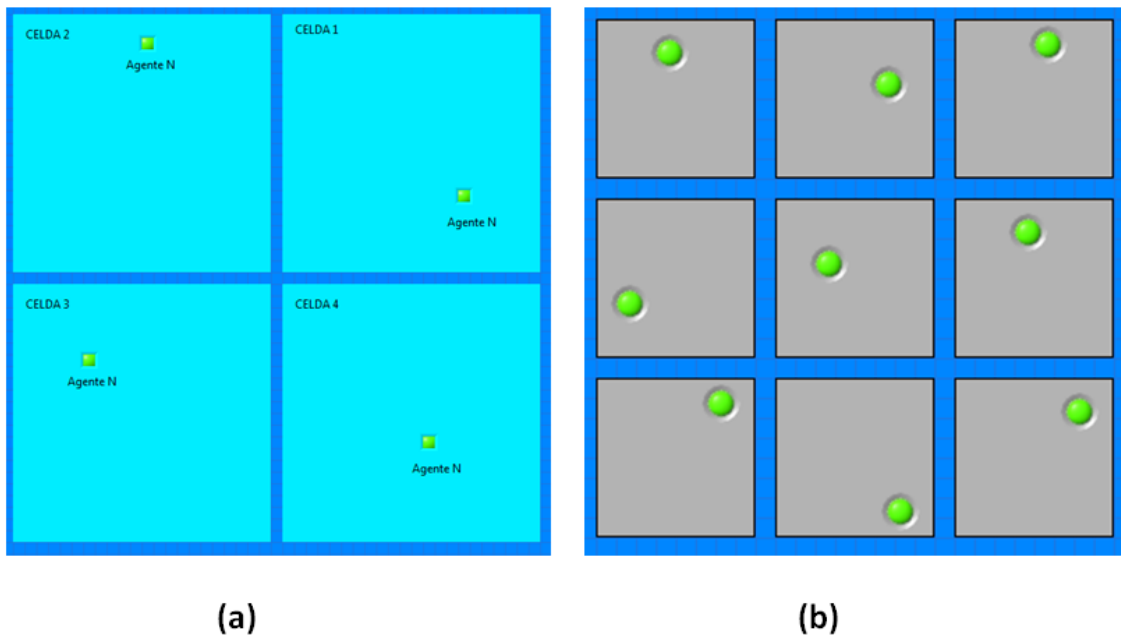


**Figura 7.2 Dirección del movimiento de la estrategia de búsqueda del robot.**

Este movimiento lo ejecutará permanentemente (siempre y cuando tenga carga suficiente en la batería) si no ha encontrado una fuente de odorante en su superficie. Notar que esta superficie corresponde a la celda donde interactúa un solo agente, por defecto el sistema de supervisión ha contemplado cuatro agentes pero de hecho en la estrategia pueden existir  $n$  agentes en  $n$  celdas diferentes que cubrirán superficies de inspección muy



grandes, tal como se puede apreciar en la Figura 7.3 para cuatro agentes y para  $n$  celdas con  $n$  agentes.



**Figura 7.3 (a) cuatro agentes (b)  $n$  agentes,  $n$  celdas.**

Esto se ha conseguido definiendo una sola estrategia de movimiento y giro conjuntas y que se repiten cuatro veces, es decir un lazo for de 4 iteraciones, anidado en otro lazo for que hace diez veces el movimiento pero que en cada iteración reduce su alcance máximo, reduciendo la distancia. Originalmente se hacía que el movimiento fuera en espiral, haciendo que el robot terminara su camino en el centro del cuadrado y luego de lo cual debería regresar a la coordenada (0,0) para volver a comenzar su trabajo idealmente infinito, pero como se verá en la sección de resultados, el error introducido en el movimiento del robot es muy grande y acumulativo, hacía que el robot no regresará nunca a su posición original; este error hacía que en la condición “infinita” de su operación sea muy poco predecible el éxito del robot para el tiempo  $n$ , no tanto como para encontrar la fuente de odorante por sí mismo, tanto como para informar al sistema de supervisión y a sus compañeros la localización de éste. Además de que las coordenadas del robot son fácilmente medibles porque en cada movimiento (de los 4 posibles) una de las coordenadas es constante y la otra varía (en el movimiento inicial)  $X$  es constante y  $Y$  varía; y estos valores se conocen directamente porque son los índices que transcurren en la iteración de cada lazo for.

Un detalle importante del sistema de  $n$  agentes y  $n$  celdas, será hacer que cada agente comience (en el tiempo “transitorio”), su trayectoria en momentos aleatorios distintos de

tal forma que en el tiempo permanente se vigile la mayor cantidad de espacios posibles y sea un solo agente a la vez el que detecte la fuente de olor.

Para llevar a cabo esta estrategia se planteó inicialmente la idea de generar la trayectoria manejando la velocidad de los motores por PWM y la distancia recorrida por tiempo de funcionamiento de los mismos, pero existía un error bastante agudo en los giros y las trayectorias, ya que a pesar de ser los mismos dispositivos conectados en las dos ruedas (hablamos de motores, ruedas y encoders) la velocidad y movimientos de ambas ruedas no son definitivamente iguales. Este error se produce también al controlar la estrategia de la forma eficiente.

Al tratar de implementar esta estrategia se evidenció la poca estabilidad de la misma por dos razones fundamentales:

- (a) Los motores a pesar de tener el mismo ciclo de trabajo y frecuencia en la señal de alimentación PWM al driver, las mismas ruedas y aros, y los mismos encoders, no giraban a la misma velocidad, lo que hacía que en la trayectoria “recta” el robot tenga una pequeña o grande curvatura, que impedía que se realice la misma.
- (b) Los motores se ponen en funcionamiento de forma idealmente infinita, hay que recordar que no son motores paso a paso que se puedan controlar por ángulos de giro u otra forma, son motores cc que funcionan por las polaridades que tienen sus terminales y giran uno u otro sentido de acuerdo a ellas, por esta razón se tiene que hacer funcionar el motor y hacerlo detener en un tiempo preciso para que describa un movimiento de x trayectoria, en muchos casos estos retardos, introducidos por software, no eran los adecuados, y no se producían para los mismos valores en iguales movimientos, haciendo que el error crezca.

A pesar de que se fueron obteniendo mejores resultados con la programación de retardos personalizados, es decir lazos anidados de diferentes valores los errores se seguían produciendo y o se podía encontrar los valores adecuado para lograr eficiencia en el movimiento.

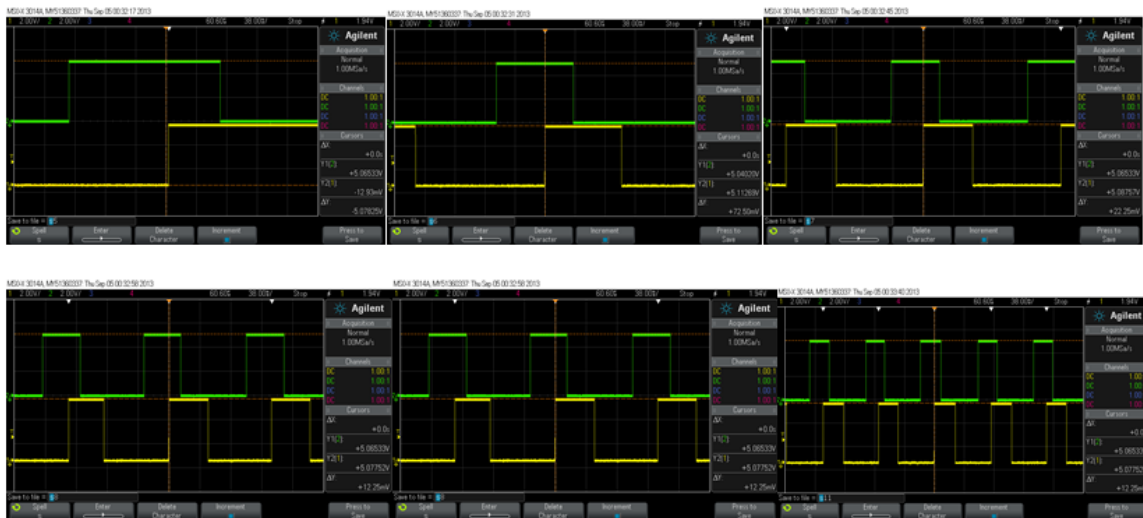
Es por esta razón que se ha decidido utilizar todos los recursos que posee la plataforma y hacer que los movimientos del robot respondan a los giros de las ruedas, independientemente de la velocidad que ellas tengan, es decir que el robot avance una distancia x definida por los encoders que tienen las ruedas y lo hagan exactamente en cualquier velocidad, en tal caso se demorará más o menos tiempo en llegar a su distancia límite.

Lamentablemente no existe realmente un verdadero sistema de control de los motores, que permita regular la velocidad de ambas ruedas y así la del robot en general, pues al ser

un sistema de dos ruedas independientes, para tener el control absoluto del movimiento se debería poseer dicho control de los motores, haciendo a lo mejor un algoritmo de control realimentado para las salidas PWM que varíen en el tiempo conforme se presente un error en el movimiento.

Al aplicar la lectura de los encoders se logró reducir el error de desplazamiento del robot, pero jamás se logró eliminarlo.

Para este conjunto ruedas, motores y encoders se ha utilizado lo siguiente: salidas PWM individuales para los motores, configuradas de forma idéntica, en frecuencia y relación de trabajo (a pesar de que se tuvo que hacer variaciones empíricas para lograr reducir el error), la lectura de los dos encoders de una sola rueda (por facilidad del algoritmo) y porque las ruedas no se mueven de forma igual y se desplazan entre ellas haciendo que las ranuras y los dientes en cada una no coincidan y se introduzcan errores. Para ellos se analizó el desplazamiento que tienen los encoders de las ruedas para cada frecuencia configurada y para relaciones de trabajo diferentes, los resultados vistos en el osciloscopio Agilent 3543 son los que se muestran en la Figura 7.4.



**Figura 7.4 Medición de la respuesta de los encoders de las redes Pololu por diferentes relaciones de trabajo.**

Como se observa para cualquier frecuencia de la señal PWM y cualquier relación de trabajo (llamada duty\_percent) el desfase de las señales entre los encoders es de  $90^\circ$ , lo que nos ayuda a verificar la posición de la rueda y la distancia que recorre para cada vuelta de la rueda y más aún para cada pulso producido por un cambio de diente a ranura y viceversa.

Otra funcionalidad que se puede observar es que la señal del encoder A, señal verde se adelanta a la Amarilla (encoder B) si la rueda gira en sentido horario y al revés, es decir la señal amarilla se adelanta a la verde, gira en sentido antihorario.

Esta funcionalidad sirve para saber y detectar cuando el robot está retrocediendo, o girando, pero no se la ha utilizado porque los movimientos del robot son absolutamente controlados y no existe la posibilidad de que se pierdan para detectar cuál es su verdadero desplazamiento.

Las funciones de software que se han creado en base a todo lo anteriormente mencionado del hardware son las siguientes:

- **int confENCODER();**
  - Crea los puertos de entrada de los encoders signados como gpio1\_6 y gpio1\_7, es decir gpio38 y gpio39.
  - Cambia la dirección de ambos gpio a modo de entrada, para poder leer los valores digitales entregados por el buffer (recordar que la señal no viene directamente del encoder, porque este entrega una señal de 0 a 5 Vdc y el GPIO de Beaglebone solamente soporta 3.3 Vdc.
- **int leerENCODER();**
  - Lee los subdirectorios del kernel de Linux ubuntu: /sys/class/gpio/gpio38/value y /sys/class/gpio/gpio39/value para saber los estados que se están produciendo y cuantos pulsos hay que contar.
- **int closePIN();**
  - borra los pines gpio38 y gpio39 de la lista de GPIO disponibles en omap\_mux.
- **int motor1F();**
  - Configura los pines gpmc\_a2 y gpmc\_a3 como ehrpwm.1:0 y ehrpwm.1:1 respectivamente.
  - Reserva las señales arriba mencionadas para el uso de PWM.
  - Resetea la relación de trabajo de la señal PWM (requisito indispensable para poder cambiarla a ella y a la frecuencia).
  - Configura la frecuencia de trabajo (1 nS).
  - Configura el ciclo de trabajo de forma personalizada. Estos valores se han obtenido experimentalmente desde un valor mínimo de 60% hasta un valor máximo de 99%, no se puede bajar menos de 60% porque el valor medio de la tensión de entrada no alcanza a arrancar los motores.
  - Se pone a correr las salidas PWM.
- **int stop1F();**
  - Para la ejecución de la salida PWM en los puertos ehrpwm.1:X.

- **int motor2F(); int stop2F();**
  - Hacen lo mismo para las señales mcasep0\_aclkx y mcasep0\_fsx signados como ehpwm.0:0 y ehpwm.0:1 respectivamente.
- **int motor1R(); int motor2R(), stop1R(), stop2R();**
  - Hacen la misma configuración anterior solo que cambiando la señal PWM, según la tabla 4.9.
- **int movimiento(int);**
  - Ejecuta el movimiento cíclico que se realiza una vez por cada lado del cuadrado de la celda correspondiente.
  - Controla los valores de los índices que rigen el movimiento.
  - Cuenta y compara los pulsos que lee el encoder a través de la función **int retardo(int);**
  - La función retardo(int) internamente llama a la función **int leerENCODER();** que lee mediante un archivo el contenido de los gpio configurados de entrada de los encoders; el parámetro que recibe es el de la distancia.
  - Huele el olor a través de la función **int detección();**
  - Envía las señales de comunicación a través de la función **int comunicaciones();**
  - El parámetro que recibe esta función está determinado por la distancia que se quiere recorra el robot, es decir está diseñado de tal forma que factor sea el número de pulsos que leen los encoders, y como cada pulso fue trasladado a distancia, es indirectamente la medida de la distancia que se desea recorra el robot.
  - Dentro de retardo se lee también el valor de los ADC en **int readADC();** y se asignan a variables de orientación.
- **int cerrarCONF();**
  - Es la función que deshabilita todos los pines del header y los vuelve a su valor por defecto; manda a stop a las señales PWM, vuelve a modo inicial a los puertos GPIO, y deshabilita la función de transmisión de UART.
  - Con las entradas analógicas no se puede hacer nada, de tener polarización la placa éstas seguirán entregando el valor digital correspondiente al voltaje de entrada que posean, sea desde los sensores de luz, como de los sensores de distancia.
- **int retardo(int);**
  - Recibe un parámetro numérico que en la práctica es el valor de la distancia que se desea recorra el robot, pues es el valor que se compara en esta función con los pulsos enviados por los encoders, mientras el valor de estos sea menor que el especificado en su parámetro los motores siguen funcionando con run en PWM, cuando ha terminado se los para con stop en PWM.

- Dentro de esta función también se “huele”, es decir se llama a la función denominada **int deteccion()**.
- Aquí también se invoca a la función **int comunicaciones()**, en donde se envía la información de las coordenadas del robot.
- **int detección();**
  - Detiene los motores instantáneamente para indicar que se ha detectado la fuente de olor y que se pasa a un caso excepcional del programa (de hecho al final de esta función se termina el programa principal de forma abrupta y el robot reaccionará nuevamente cuando se le haga reset y coloque manualmente en la posición (0,0)).
  - Investiga con secuencias de condicionales cuál, o cuáles sensores fueron activados y los detecta, al mismo tiempo asigna a cada uno de esto un número de caso particular, por ejemplo el sensor norte valor 1; sensores norte y este valor 2, etc.
  - Ejecuta una acción de movimiento para cada uno de los casos antes asignados. Esta acción consiste en detectar la línea perpendicular a la dirección de la fuente de odorante, girar 90° en sentido de ella y activar el sensor principal (norte) y hacerlo “oler” hasta que este llegue a un umbral máximo, momento en que se detiene, envía la posición repetidamente para que el sistema de supervisión tome la acción correctiva.
  - El sistema de supervisión tendrá que cambiar el color del indicador del robot de verde a rojo y tomar la acción de comunicación a los demás agentes y lo que se considere norma de procedimiento en el en su manual de operador.
  - Finalizado la función detiene todas las configuraciones de los puertos y pines del header que el robot utiliza (hace un reset de todo) y sale del programa principal.
  - El robot no se apaga porque la batería sigue funcionando, pero será necesario en el manual de procedimientos del operador humano que se indique que debe correr nuevamente el programa cuando ponga en acción al robot nuevamente.
- **int comunicaciones();**
  - En primer lugar utiliza una función denominada **int confCOM()**, que lleva los pines a UART1\_RXD y UART1\_TXD a modo 0, es decir a opción de uso de uart.
  - Luego envía el dato de posición a través de la escritura del dato como un número decimal de las coordenadas de X e Y que son obtenidas realmente de los índices de los lazos de la función retardo y que se han quedado estáticos el momento que retardo envió a la función detección al programa.

- El entramado de esta información solamente consiste en formar un número de dos cifras con los valores de X e Y, multiplicando el uno por 10 y sumándolo el otro.

Todas las funciones y el programa principal, ha sido desarrolladas en lenguaje C++ y compiladas en el kernel de Linux Ubuntu interno de la placa Beaglebone con la instrucción: **c++ programa.c -o programa**, que a partir del código fuente **programa.c** crea el código objeto ejecutable **programa**, este es el que se corre en el kernel de Linux con opción base con la instrucción: **root@beaglebone:~ ./programa**.

## 7.3 Pruebas y resultados.

Las pruebas realizadas se basan en la necesidad de comprobación que los sistemas elaborados y los concebidos idealmente funciones de forma correcta y las correcciones que se deben hacer de no hacerlo, los errores que en forma práctica se detectaron en cada uno de los sistemas individuales del robot y finalmente en las tareas generales desarrolladas por el robot, se presentan además los resultados de dichas pruebas.

### 7.3.1 Prueba de motores.

Para probar que los motores funcionaban adecuadamente a los requerimientos se debe variar las características de las señales de alimentación a él, como se pudo observar en el diseño del robot y su construcción, las señales de salida de los puertos PWM de la placa Beaglebone no se alimentan directamente a las entradas + y – del motor, sino que son previamente “pasadas” por un driver, que da la posibilidad de manejar niveles de potencia mayores que los que haría la salida PWM de Beaglebone. Para ello se realizaron dos pruebas.

1. Se alimentó a los motores directamente desde la señal de salida PWM; el resultado no es ni siquiera registrable, el movimiento de los motores era tan pobre que respondía a valores altos del valor medio de la señal digital de salida, por lo que se descartó definitivamente ya que tenía poquísimas opciones de regulación.
2. Se alimentó a los motores de acuerdo a las alternativas de conexión sugerido por el fabricante del driver, ver tabla 4.10, y se realizaron pruebas de funcionamiento para determinar el umbral de funcionamiento de las señales PWM. Los resultados se visualizan en la Tabla 7.1.

**Tabla 7.1 Apreciación del movimiento de los motores a variaciones de la señal PWM**

duty_percent de señal PWM (%)	Movimiento del motor	Tracción de rueda
10	Nulo	Nulo
20	Nulo	Nulo
30	Nulo	Nulo
40	Nulo	Nulo
50	Nulo	Nulo
60	Minimo	Nulo
70	Aceptable	Aceptable
80	Muy aceptable	Muy aceptable
90	Muy aceptable	Muy aceptable
99	Muy aceptable	Muy aceptable

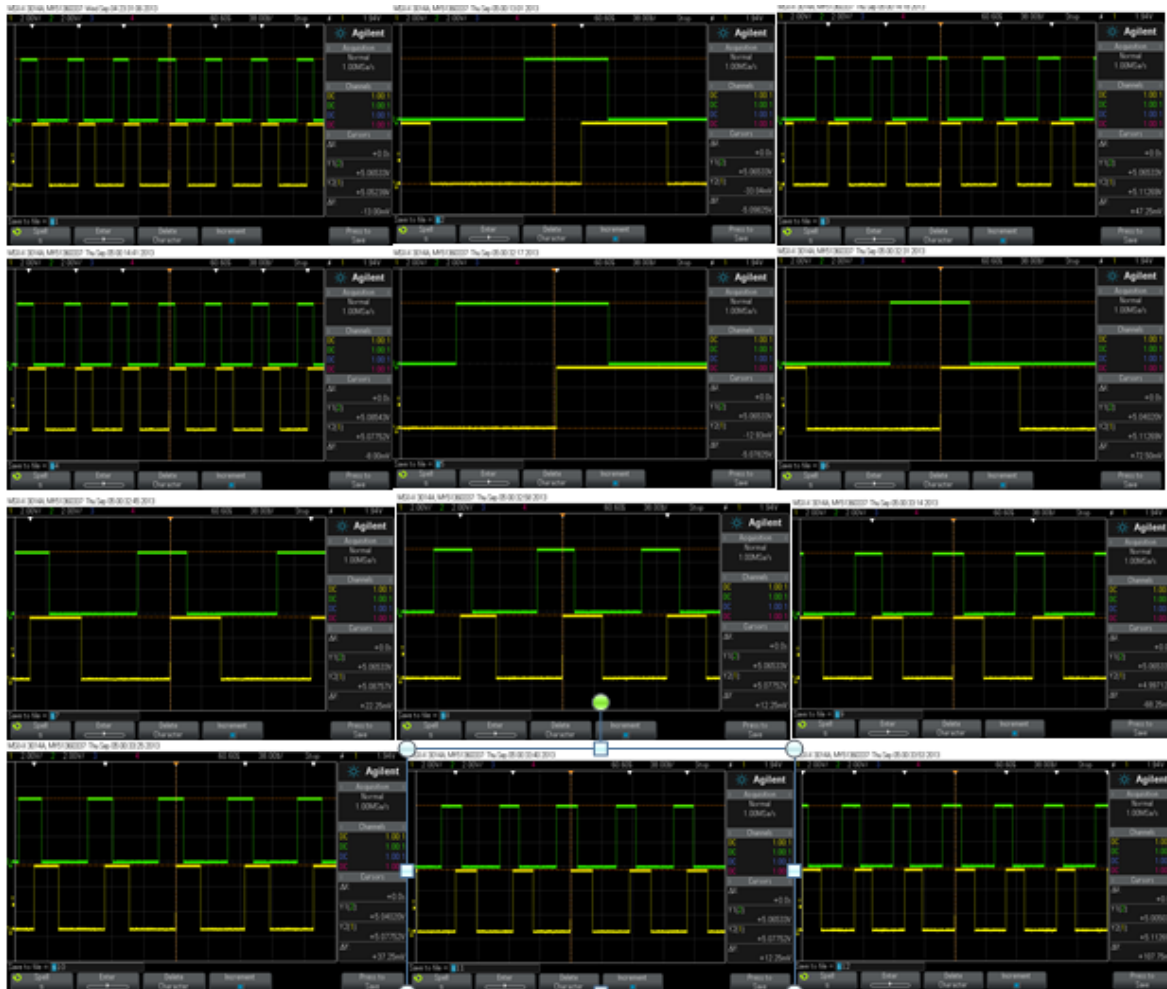
De lo cual se observa que la variación apreciable es desde 60% en adelante. Experimentalmente se determinó que la velocidad del robot a 60% de relación de trabajo del PWM era casi nula, porque no tenía el motor el torque suficiente para mover todo el cuerpo del robot, a pesar de que en “vacío” si las movía. La real variación experimental del robot se obtuvo desde 61% al 99% apreciando los cambios en los movimientos del mismo. Situación que se aclara, porque los datos que se presentan también fueron obtenidos con una frecuencia de la señal de PWM de 1 KHz, el motor ha respondido pero las variaciones del movimiento del robot han sido imperceptibles en intervalos cortos del orden de las decenas, es decir no se aprecia cambio de velocidad entre 65, 70, 75, 80% de relación de trabajo, menos aún 85, 90 y 99%. Es por esta razón que los valores escogidos son para la velocidad media del robot son:

- Frecuencia de la señal PWM: 1MHz.
- Relación de trabajo PWM: 70 – 85%.

### **7.3.2 Lectura de encoders.**

En el caso de la lectura de los encoders de las ruedas, la necesidad estaba en el hecho de comprobar que se mantenga la relación de desfase de 90° entre los encoders, y para las frecuencias sobre las cuales se estaba tratando de operar la señal PWM. Los resultados fueron 100% efectivos. Para todas las mediciones de frecuencia la relación entre la señal de salida de los encoders A y B fue de 90°. Esto se puede apreciar en el conjunto de figuras que se presentan en la Figura 7.5.





**Figura 7.5 Conjunto de pruebas de respuestas de encoder A y B para diferentes relaciones de trabajo y frecuencia.**

Un aspecto a tomar muy en cuenta en esta prueba es la lectura de los valores de los encoders pues la velocidad de variación de las señales PWM máxima es de 1 MHz, pero las iteraciones del programa se pueden realizar a la velocidad del procesador de forma ideal, es decir se tuvo que hacer un control por software porque la velocidad de lectura del procesador era infinitamente mayor que la velocidad de cambio de los pulsos, esta estrategia de software sirvió para determinar los verdaderos pulsos y no leer estados anteriores o intermedios de los dientes o ranuras de las ruedas.

### 7.3.3 Lectura del ADC.

Las entradas analógicas de la placa Beaglebone están destinadas en el diseño del robot tanto para la lectura de los sensores de distancia de obstáculos, originalmente, como para la lectura adaptada de los sensores infrarrojos de la fuente de luz, es decir a la simulación de la nariz.

La primera opción no se ha implementado por software, a pesar de que el hardware está disponible, es solamente conectar los sensores y hacer las funciones correspondientes, porque se probó su funcionamiento y está a punto, solamente utilizar un sensor adecuado que emita los valores máximos y mínimos de los obstáculos y se puede implementar.

La segunda opción si se probó y arrojó resultados en varias pruebas, es así que los sensores utilizados hicieron que la información disponible cambiara los diseños del hardware, es así que se cambiaron los valores de las resistencias en la fuente de luz y en los detectores para mayor resolución de la distancia de detección mínima del sensor. Los resultados se observan en la Tabla 7.2 y Tabla 7.3.

**Tabla 7.2 Valores para el emisor de luz OP293B**

Resistencia Serie del detector	Palabra digital producida en la distancia de:					
	5cm	10cm	20cm	30cm	50cm	80cm
1K $\Omega$	80	0	0	0	0	0
10K $\Omega$	1232	85	85	85	85	85
100K $\Omega$	2340	967	490	490	490	490

**Tabla 7.3 Valores para el emisor de luz SFH 415**

Resistencia Serie del detector	Palabra digital producida en la distancia de:					
	5cm	10cm	20cm	30cm	50cm	80cm
1K $\Omega$	-	-	-	-	-	-
10K $\Omega$	-	-	-	-	-	-
100K $\Omega$	3590	3045	2677	2321	1980	1450

Se debe aclarar que como la mejor respuesta se obtuvo con la resistencia en serie del sensor de 100K $\Omega$ , y por la limitación del tiempo ya no se cambió el hardware y solamente se hizo la prueba con ella.

Por otro lado esta tabla verifica que la detección mejoró enormemente inclusive con la misma corriente de 100mA por cada led emisor, la prueba del cortocircuito, echó a perder el mosfet y no se verificó la distancia de recepción, afortunadamente esto solamente se realizó en un protoboard.

### 7.3.4 Comunicaciones.

Para las comunicaciones se debían hacer tres pruebas, la primera de configuración y envío de los mensajes entre los módulos XBee con sus respectivos parámetros, realizada con X-CTU o HYPERTERMINAL, la otra del envío de los datos desde la plataforma robótica y su lectura en el módulo de USB explorer y finalmente el envío de datos del robot y su lectura en el sistema de supervisión ya configurado.

La primera prueba arrojó los resultados esperados, solamente se escribieron los parámetros mencionados en la sección 6.2. en los módulos, se descargó y ejecutó X-CTU, y se arrancó el terminal, transmitiendo y recibiendo en forma bidireccional los datos en un porcentaje de logro del 100%, es decir cada vez que se transmitió se recibió el dato en el módulo contrario.

La segunda prueba consistía en el envío de los datos desde el módulo XBee conectado en el robot a su coordinador conectado en el USB explorer en X-CTU, esta prueba ya no tuvo el porcentaje de éxito que la anterior pues en primer lugar se lo hizo de dos formas:

1. Colocando datos en el kernel de Linux Ubuntu de forma manual, a lo que el modulo coordinador recibió los datos 10 veces de 10 pruebas, es decir con una eficacia del 100%.
2. Realizando un programa dedicado en la Beaglebone que enviaba datos al archivo de salida del UART y luego por el módulo. Esta prueba no tuvo tanto éxito porque la velocidad de envío de los datos por la Beaglebone es demasiado rápida en comparación con el sistema de comunicaciones, entonces se le puso un retardo de 1 seg. (producido por una instrucción sleep) y se recogían los datos de forma adecuada, es decir allí si el módulo los leía en un 100%. Pero se producían dudas de que este retardo se ejecute en el programa principal como así fue.
3. Se leyeron los datos que mandaba el robot de forma manual y sin problemas usando la técnica anterior, pero si se quitaban los retardos se volvían a perder los datos. Pero se acepta estas pérdidas ya que los sistemas de supervisión de este tipo no manejan tiempo real estricto, pues el operador humano, no puede contextualizar la rapidez de estas variaciones y no las detecta y el sistema no las puede presentar porque pierde su intuitividad. Entonces se decide que los datos tranquilamente se pueden observar cada 1 segundo en el sistema de supervisión, pero en el de envío no se pueden retardar ese tiempo, porque la lectura y escritura de datos en el robot se hace siempre que se leen los encoders y eso se hace en forma continua en el programa. Entonces se acepta que los datos se pierdan y el sistema los lea desactualizados, ya que como son los mismos aunque se pierdan en la siguiente iteración del programa de supervisión se vuelven a repetir, y de nos ser

así el error no es tan grande porque los datos son muy contiguos y lo máximo que se puede perder es de 1 a 2cm que es lo que gira la rueda del robot en esos intervalos, eso se deberá considerar en el sistema de supervisión a lo mejor realizando un promedio de las ultimas 10 muestras y aceptando o no los valores posteriores y de ser el caso interpolándolos.

### **7.3.5 Pruebas del robot.**

Para el robot en general se han realizado dos pruebas fundamentales: (1) pruebas de exactitud del movimiento y (2) pruebas de exactitud de olor.

#### **7.3.5.1 Pruebas de error en el movimiento.**

Como se dijo anteriormente, al no haber un sistema de control del movimiento del robot, se está únicamente a la respuesta individual de los motores de cada rueda.

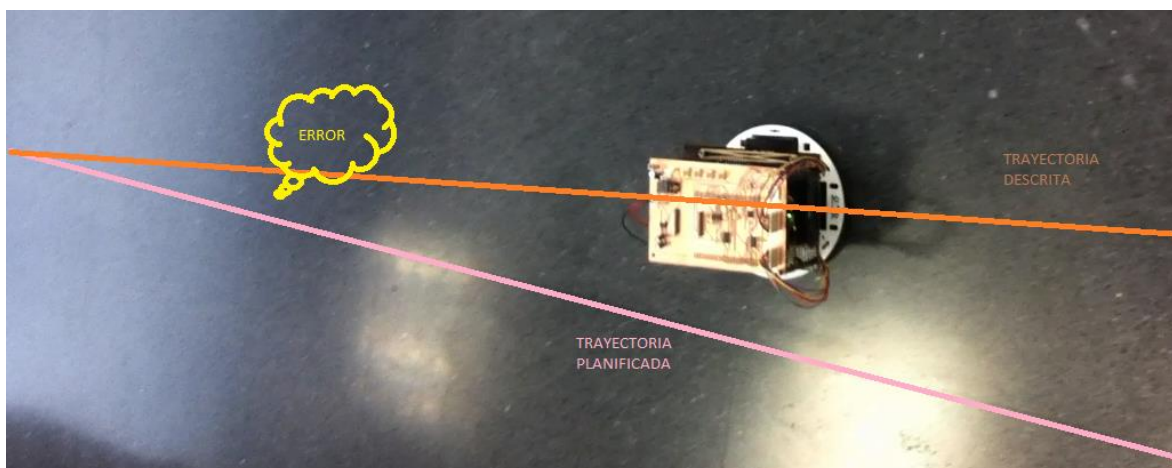
Como explique en las secciones anteriores, los motores han sido programados exactamente de la misma forma, es decir en frecuencia y porcentaje de trabajo, para las cuatro salidas PWM, dos para el motor izquierdo y dos para el derecho. A pesar de eso y como se debía suponer solamente requisito, este necesario para el movimiento de las ruedas, no es así. Porque cuando se mueve el robot hacia adelante empieza en cada movimiento a introducir un error y a tener cierta curvatura, que significa que en la práctica la una rueda es más rápida que la otra, además que paulatinamente va introduciendo más error aun, pues las ruedas no salen alineadas unas de otras y la lectura de los encoders se distorsiona en cada arranque, además de que cuando arranca con porcentajes altos de PWM, gira un poco el robot inicialmente e introduce error desde el inicio.

Estos errores se van acumulando para cada “vuelta” que da el robot en su celda de supervisión, lo que hace que no recorra adecuadamente dicha superficie. La solución más practica poner en ejecución un sistema de control realimentado para obtener exactitud y precisión en los movimientos del robot. Trabajo largo y complicado que no se ha hecho.

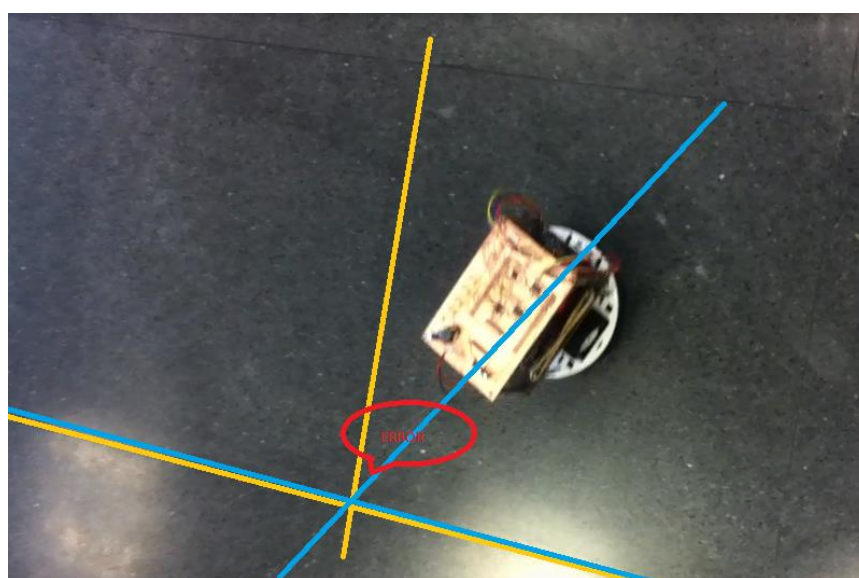
La solución encontrada que no ha solucionado al 100% el problema consistió en variar en forma empírica los porcentajes de duty\_percent en ambas salidas PWM de los dos motores hasta hacer que se reduzca al máximo la variación de la trayectoria, y reducir las trayectorias rectas que tienen los agentes, haciendo idealmente el número de celdas más grande.

Este error también introduce cambios en los ángulos de giro del motor es por eso que en las instancias finales gira más de los grados que los hace en las iniciales para la misma trayectoria. A pesar de ello, el prototipo ha funcionado exitosamente un alto porcentaje

de las veces a las que se ha puesto a prueba, esto se puede visualizar en la Figura 7.6(a) y (b) y la Tabla 7.4.



**Figura 7.6 (a) Error introducido en la trayectoria del robot.**



**Figura 7.6. (b) Error introducido en el ángulo de giro.**

**Tabla 7.4 Pruebas realizadas al movimiento “recto” del robot y sus aciertos.**

N. Prueba	1	2	3	4	5	6	7	8	9	10
desviación	40cm	30cm	30cm	20cm	16cm	18cm	15cm	10cm	8cm	6cm
Angulo	30°	28°	25°	24°	22°	12°	10°	5°	5°	2°

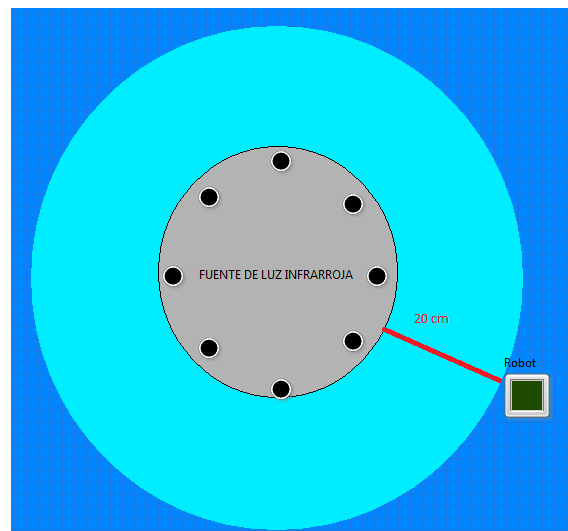
De estos resultados se puede apreciar que con las calibraciones y los tanteos que se realizaron en los porcentajes del duty\_percent y la frecuencia de los motores se ha

reducido el error pero persiste en un 4% en la trayectoria recta y en el ángulo, que mejoró mucho más porque la velocidad de giro es más lenta es de 3% más o menos.

Experimentalmente se comprueba que hay menos error en el ángulo de giro que en la trayectoria recta; y mientras más larga más error.

### 7.3.5.2 Pruebas del error en la detección de la posición de la fuente de olor.

Para las pruebas de búsqueda de la fuente de olor se han obtenido mejores resultados, porque se considera que la tolerancia máxima para aceptar por encontrada y localizada la fuente de error es 10%, como la distancia máxima a la que se encuentran los agentes de ella sería un lado del cuadrado es decir 2 metros, debería estacionarse a una distancia de 20cm del centro de la fuente de olor. De acuerdo como se indica en la Figura 7.7



**Figura 7.7 Error permitido para detección de la fuente de luz infrarroja.**

Experimentalmente se comprobó que el 90% de las veces el robot estuvo a una distancia dentro del error permitido. Tabla 7.5.

**Tabla 7.5 Distancia del robot a la fuente de luz en pruebas.**

Prueba	Distancia robot fuente de olor
P1	5cm
P2	12cm
P3	15cm
P4	10cm
P5	10cm
P6	1cm

P7	10cm
P8	10cm
P9	4cm
P10	30cm
P11	26cm
P12	12cm
P13	7cm
P14	2cm
P15	3cm
P16	2cm
P17	4cm
P18	4cm
P19	6cm
P20	2cm

Por lo que se puede observar que la mayoría de las pruebas han caído en el rango especificado. Esto se da porque sea el error que se haya introducido en la trayectoria total del robot, es un programa a parte el de encontrar y trasladarse a la fuente de olor y siempre tiene similares características, entonces los resultados son muy claros.





## **CAPITULO VIII**

### **COCLUSIONES Y TRABAJO FUTURO**



## 8 Conclusiones y trabajo futuro.

### 8.1 Conclusiones.

En el presente trabajo se ha desarrollado una plataforma robótica inteligente en el sentido de autonomía de movimiento y localización, que cuenta con la característica más eficiente de desplazamiento y movilidad, así como facilidad de identificación de las distancias que recorre y los ángulos de giro que efectúa, en él se ha integrado un simulador de nariz electrónica constituido por cuatro sensores de luz infrarroja, que es el fenómeno más parecido al comportamiento de odorantes, este agente robótico, tiene la inteligencia suficiente para localizar y dirigirse a la fuente de luz que ha detectado, y una vez llegada a la posición donde ésta está, comunicar mediante un sistema de red de comunicaciones inalámbricas Zigbee la posición de ésta fuente del fenómeno, a un sistema de supervisión centralizado, que colaborará con los demás agentes para que puedan desarrollar el trabajo de movilización hacia la fuente del fenómeno. Este sistema está desarrollado por un software de alto nivel en un ordenador que actuará de forma que pueda colaborar eficientemente tanto a nivel local como a nivel global.

Se ha llegado a la conclusión que el tipo de robot que mejor se desempeña es el de plataforma idéntica al robot utilizado en [6], pero mejorado en aspectos de capacidad computacional, facilidad de programación y comunicaciones; además de que los elementos con los que cuenta el robot desarrollado, le permiten localizarse plenamente en el ambiente en que va a interactuar, siendo éste aspecto solo una necesidad actual del sistema ya que puede extenderse con modificaciones mínimas del hardware y creación de funciones de software con otros sistemas de comunicaciones de mayor envergadura.

Se ha simulado, he instalado un sistema de detección de un fenómeno físico que más se aproxima al comportamiento de los olores, pero se ha implementado el hardware necesario para hacer que se adapte en cualquier momento y bajo cualquier necesidad una nariz electrónica inteligente. Este sistema de simulación sigue un algoritmo de detección que puede ser utilizado con mínimas modificaciones para dicha nariz electrónica.

Se ha construido un hardware electrónico que acopla de manera completamente eficiente a todos los componentes de diferentes plataformas y constructores de tecnología propietario, que han intervenido en el prototipo, y este se ha realizado con tecnología moderna.

Se ha escogido, configurado e implementado un sistema de comunicaciones punto a punto que con modificaciones mínimas se puede convertir en punto multipunto para implementar una red de agentes que efectúen su operación en un gran número de celdas,

y que se conecta con una plataforma de supervisión que puede escalarse para producir diversos niveles de supervisión, esta plataforma básica está programada pero se puede escalar mucho más por la capacidad y efectividad del software empleado. Y que desempeña un trabajo colaborativo eficaz entre agentes del mismo tipo.

En el aspecto personal se dio un salto generacional al descubrir y aplicar la ciencia y la técnica de vanguardia en todos los aspectos del análisis diseño e implementación del prototipo.

## **8.2 Trabajo futuro.**

El robot construido puede tener perfeccionamiento futuro en los siguientes aspectos:

Diseñar e implementar un sistema de control del mecanismo motor del robot, un sistema realimentado que haga que se desplace con error por las trayectorias elegidas y que de esta forma pueda más efectivamente informar a los demás agentes las posiciones en las que se encuentra él y la fuente de odorante.

Implementar un sistema de visualización de los movimientos y localizaciones de este robot y los demás agentes del sistema, por la incorporación de una pantalla táctil en la plataforma Beaglebone, que mantiene la mayoría de los elementos necesarios para esto desocupados.

Implementar sistemas de comunicaciones con robots de mayor envergadura y automóviles ya que la red CAN BUS que dispone la plataforma Beaglebone es una red establecida para sistemas automotrices, entonces se puede realizar un intento de adaptación de los movimientos de este robot a vehículos más grandes.

Implementar redes de tipo Punto a multipunto con la misma plataforma y modificando la misma con nuevos módulos XBee crear redes MESH que darán versatilidad al trabajo colaborativo.

Desarrollar totalmente el sistema de colaboración del sistema de supervisión de tal manera que se controle al 100% la red y los servicios que pueden prestar cada nodo y definan acciones especiales o tareas alternativas para cada uno de ellos, además que se podría conectar a sistemas de más alto nivel como redes profibus, profinet, Ethernet Industrial, etc.



# Referencias.

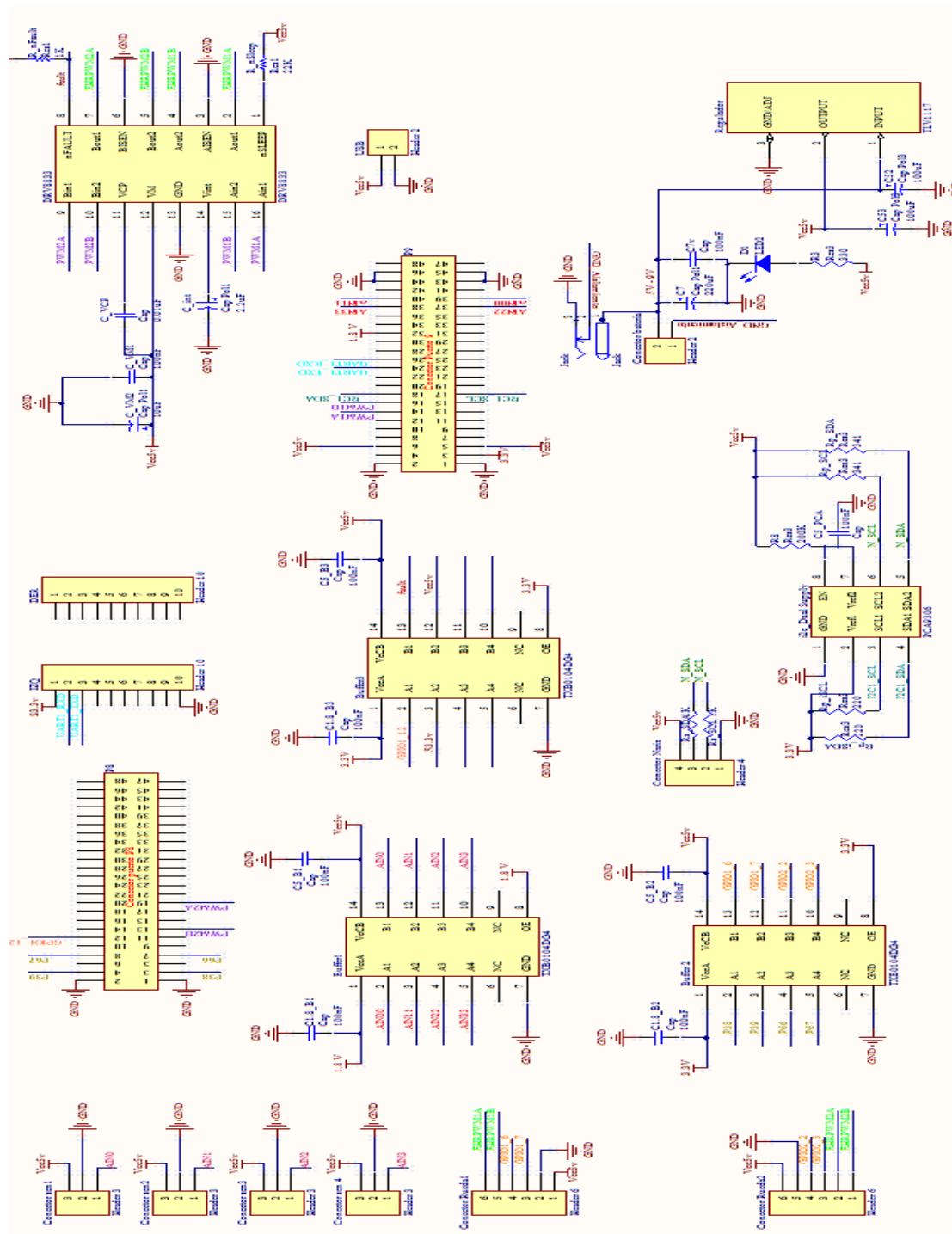
---

- [1]. Khorasgan Branch, Isfahan, Iran (2009). Electronic Nose Technology and its applications. *Sensors & Transducers Journal*. 19-25.
- [2]. Wilson, A. D., & Baietto, M. (2009). Applications and Advances in Electronic-Nose Technologies. *Sensors*, 9(7), 5099-5148. doi:10.3390/s90705099.
- [3]. Morsi, I. (2008). Electronic Noses for Monitoring Environmental Pollution and Building Regression Model. *Science And Technology*.
- [4]. Song, K., Liu, Q., & Wang, Q. (2011). Olfaction and hearing based mobile robot navigation for odor/sound source search. *Sensors (Basel, Switzerland)*, 11(2), 2129-54. doi:10.3390/s110202129.
- [5]. Marques, L.; Nunes, U.; Almeida, A.T. Olfaction-based mobile robot navigation. *Thin Solid Films* 2002, 418, 51-58.
- [6]. José María Martín Talavera, (2012), Posicionamiento por radiofrecuencia basado en robots colaborativos. Repositorio digital, Escuela Politécnica Superior, Universidad Autónoma de Madrid. <http://biblioteca.uam.es/politecnica/default.html>
- [7]. Beaglebone System Reference Manual, Rev A.6.0.0 (2012). <http://beagleboard.org/Products/BeagleBone>.
- [8]. Tomás Vázquez Rubio, (2013), Integración de una nariz electrónica ultra-portátil en un robot modular para el control de su movimiento a través de los odorantes recibidos. Repositorio digital, Escuela Politécnica Superior, Universidad Autónoma de Madrid. <http://biblioteca.uam.es/politecnica/default.html>
- [9]. Product Manual v1.06 For OEM RF Module Part Numbers: XB24-...-001, XB24-...-002 XBP24-...-001, XBP24-...-002 ZigBee™/IEEE® 802.15.4 OEM RF Modules by MaxStream, Inc. <http://www.digi.com/support/productdetail?pid=3293&type=documentation>
- [10]. Mechanical Components » Chassis » Pololu 5" Robot Chassis RRC04A » Pololu 5" Robot Chassis RRC04A Transparent Clear. <http://www.pololu.com/catalog/product/1505>.
- [11]. Andrés Oyarce, Paúl Aguayo, Eduard Martín; Guía del Usuario; XBEE Series 1.; Revisión Julio 2010. [http://www.olimex.cl/tutorials.php?page=tut\\_xbee](http://www.olimex.cl/tutorials.php?page=tut_xbee)
- [12]. Procesador ARM CORTEX-M3, características y funcionalidades de operación; Pagina web del fabricante: ARM: The Architecture for the digital Word: <http://www.arm.com/products/processors/cortex-m/cortex-m3.php>
- [13]. <http://www.altium.com>



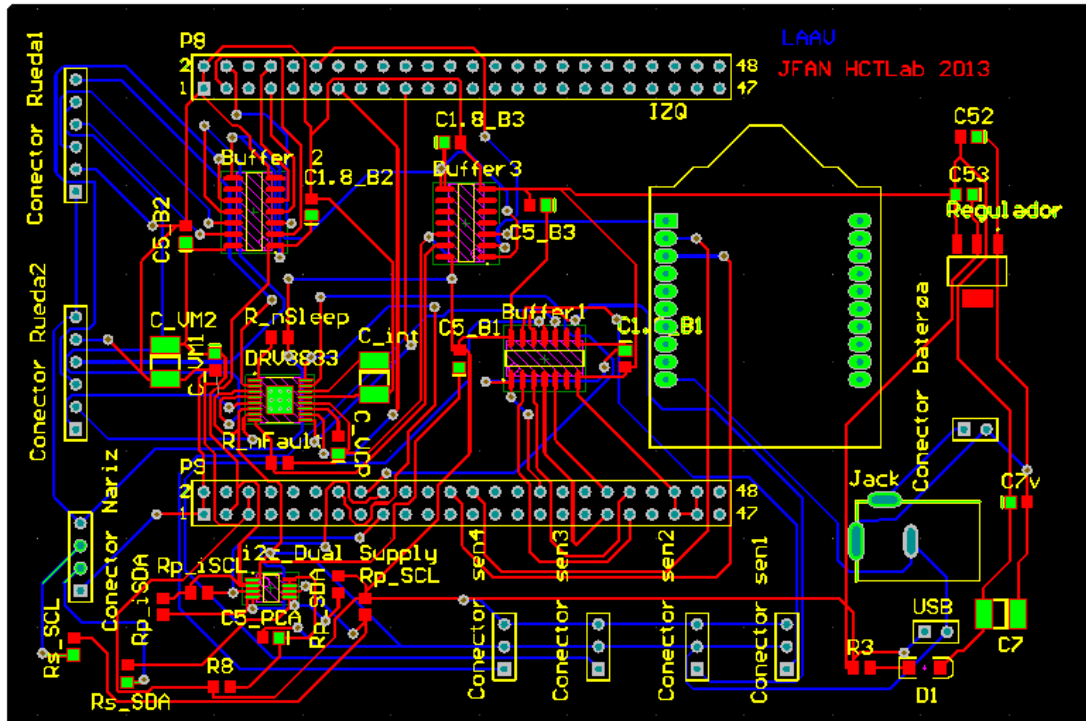
# Anexos.

## A. Diagrama Esquemático de la Placa Adaptadora Altium Designer [13]





## B. DISEÑO DEL PCB DE LAPLACA ADAPTADORA.



## C. LAYOUT DE LA PLACA ADAPTADORA.

